NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CAN SNMP BE USED TO CREATE A SILENT SS IN AN 802.16 IMPLEMENTATION?

by

Joseph K. Harrison II

September 2008

Thesis Advisor:	Rex Buddenberg
Second Reader:	Albert Barreto

Approved for public release, distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Can SNMP be Used to Create a Silent SS in an 802.16 Implementation?

5. FUNDING NUMBERS

6. AUTHOR(S) Joseph K. Harrison II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The IEEE 802.16 standard is a wireless communications standard which holds great potential for use by the U.S. military. As IEEE Std. 802.16 is a commercial standard, it can be used as a COTS solution for extending the reach of the internet down to the level of the individual soldier without incurring any development costs. Additionally, 802.16 "out of the box" supports end-to-end routing and is compatible/interoperable with other ubiquitous networking technologies such as Ethernet and IP. Given the wireless nature of 802.16, every soldier within range of an 802.16 Base Station (BS) has the potential to benefit from the flow of information from the Command and Control network, as well as the ability to contribute back to the network, increasing the situational awareness of all who are connected.

While the default configuration of 802.16 has tremendous potential, it is at its base a commercial standard. There is a potential for modification of the standard to increase the usefulness of 802.16 for the military. This thesis explores one such possibility by investigating the use of SNMP to obviate the need for a Subscriber Station (SS) to transmit, eliminating the associated risk of detection through signal tracking.

14. SUBJECT TERMS

802.16, IEEE Std. 802.16 , SNMP, Wireless

15. NUMBER OF PAGES
117

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

CAN SNMP BE USED TO CREATE A SILENT SUBSCRIBER STATION IN AN 802.16 IMPLEMENTATION?

Joseph K. Harrison II
Civilian, United States Navy
B.S., California State University Monterey Bay, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2008

Author: Joseph K. Harrison II

Approved by: Rex Buddenberg
Thesis Advisor

Albert Barreto
Second Reader

Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The IEEE 802.16 standard is a wireless communications standard that holds great potential for use by the U.S. military. As IEEE Std. 802.16 is a commercial standard, it can be used as a COTS solution for extending the reach of the internet down to the level of the individual soldier without incurring any development costs. Additionally, 802.16 “out of the box” supports end-to-end routing and is compatible/interoperable with other ubiquitous networking technologies such as Ethernet and IP. Given the wireless nature of 802.16, every soldier within range of an 802.16 Base Station (BS) has the potential to benefit from the flow of information from the Command and Control network, as well as the ability to contribute back to the network, increasing the situational awareness of all who are connected.

While the default configuration of 802.16 has tremendous potential, it is at its base, a commercial standard. There is a potential for modification of the standard to increase the usefulness of 802.16 for the military. This thesis explores one such possibility by investigating the use of SNMP to obviate the need for a Subscriber Station (SS) to transmit, eliminating the associated risk of detection through signal tracking.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

-A-

AAS	Adaptive Antenna System
AES	Advanced Encryption Standard
AK	Authentication Key
AM	Accounting Management
AM	Amplitude Modulation
AP	Access Point
ARQ	Automatic Repeat Requests
ASN.1	Abstract Syntax Notation One
ATDD	Adaptive Time Division Duplexing

-B-

BER	Basic Encoding Rules
BS	Base Station
BW	Bandwidth
BWA	Broadband Wireless Access

-C-

C2	Command and Control
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CID	Connection Identification
CINR	Carrier to Interference-plus-Noise-Ratio
CM	Configuration Management
COMSEC	Communications Security
COP	Common Operational Picture
COTS	Commercial-off-the-Shelf
CPS	Common Part Sublayer

CS	Convergence Sublayer
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access/Collision Detection

-D-

DCD	DL Channel Descriptor
DES	Data Encryption Standard
DL	Downlink
DL	Download
DOCSIS	Data Over Cable Service Interface Specification
DoD	Department of Defense
DOS	Denial of Service
DSA	Dynamic Service Addition
DSA-REQ	Dynamic Service Addition Request
DSC	Dynamic Service Change
DSD	Dynamic Service Deletion
DSSS	Direct-Sequence Spread Spectrum

-F-

FCAPS	Fault Management, Configuration Management, Accounting Management, Performance Management, Security Management
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FM	Fault Management
FM	Frequency Modulation
FOB	Forward Operations Base

-I-

IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISI	Intersymbol Interference
IT	Information Technology

-J-

JTRS	Joint Tactical Radio System

-L-

LAN	Local Area Network
LMSC	LAN/MAN Standards Committee
LOS	Line of Sight
LPI	Low Probability of Intercept

-M-

MAC	Media Access Control
MAN	Metropolitan Area Network
MIB	Management Information Block
MPDU	MAC Protocol Data Units
MSDU	MAC Service Data Units

-N-

NCW	Network Centric Warfare
NIC	Network Interface Card
NLOS	Non-Line of Sight
NMS	Network Management Station

-O-	
OFDM	Orthogonal Frequency Division Multiplex
OFDMA	Orthogonal Frequency Division Multiplex Access
OID	Object Identifier
OSI	Open Systems Interconnection
-P-	
PAR	Project Authorization Request
PDU	Protocol Data Unit
PKC	Public Key Cryptography
PKM	Privacy Key Management
PKM	Public Key Management
PKM-REQ	Privacy Key Management Request
PKM-RSP	Privacy Key Management Response
PM	Performance Management
PmP	Point to Multipoint
PtP	Point to Point
-Q-	
QoS	Quality of Service
-R-	
RFC	Request for Comment
RSSI	Received Signal Strength Indication
-S-	
SA	Security Association
SAID	Security Association ID

SAP	Service Access Point
SDU	Service Data Unit
SFID	Service Flow Identifier
SGMP	Simple Gateway Management Protocol
SM	Security Management
SMI	Structure Of Management Information
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
SS	Security Sublayer
SS	Subscriber Station

-T-

TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TEK	Traffic Encryption Key
TOS	Theft of Service
TRANSEC	Transmission security

-U-

UDC	UL Channel Descriptor
UDP	User Datagram Protocol
UL	Uplink
UL	Upload

-V-

VoIP	Voice over IP
VPN	Virtual Private Network

-W-

WiMAX	Worldwide Interoperability for Microwave Access
Wi-Fi	Wireless Fidelity
WAN	Wide Area Network

EXECUTIVE SUMMARY

In the interest of better preparing the warfighter, the Department of Defense (DoD) has moved more and more Information Technology (IT) further toward the edge of today's battlespace. More and more frequently, soldiers of all stripes are finding that IT can be used in a staggering number of combinations to accomplish or assist in an almost unlimited number of objectives. One technology that is finding its way onto the battlespace is the wireless networking standard developed by the Institute of Electrical and Electronics Engineers (IEEE), 802.16 and its accompanying technology: Worldwide Interoperability for Microwave Access (WiMAX). IEEE standard 802.16 describes a broadband wireless technology for Line of Sight (LOS) and Non-Line of Sight (NLOS) communication. As most in the IT field are familiar with IEEE Std. 802.16's cousin, IEEE Std. 802.11 (Wi-Fi), IEEE Std. 802.16 is similar in concept yet able to transport significantly more data over significantly longer distances.

In addition, where Wi-Fi is only useful at the edge of a network due to its contention based media access method, the media access method defined in IEEE Std. 802.16 is centrally managed. This central management of connections allows for a more stable network under heavy load, which allows IEEE Std. 802.16 to not only efficiently and effectively serve at the edges of a network, but for it to also play a role in the interior of the network as well as a backbone link between infrastructure and edge routers.

While this backbone capability of IEEE Std. 802.16 is valuable, this thesis will focus on IEEE Std. 802.16 at the edges of the network. One application for this type of communication is to have individual soldiers carry a wireless handheld device with which they can transmit and receive real-time Command and Control (C2) information germane to their particular battlespace. However, this new boon to the soldier does not come without a cost. The IEEE 802.16 standard is, by definition, connection-oriented. Both the transmitter and the receiver are in constant communication with each other negotiating power levels, modulation schemes, etc. While normally there is no risk associated with

these signals, there is a potential for enemy forces to use readily available tools to detect the wireless signals that an individual soldier may be transmitting and use those signals to zero in on the physical location of the soldier.

The focus of this thesis is whether this increased potential for exposure can be mitigated using the Simple Network Management Protocol (SNMP). SNMP is a networking protocol that can allow an administrator to collect data about and, more importantly for the purpose of this thesis, remotely configure networking devices. This thesis will investigate the intersection of the 802.16 standard and the SNMP protocol to determine if some conjugation of the two can be used to dynamically configure an IEEE Std. 802.16 link such that a soldier with a wireless handheld device can receive data without having to transmit back to a Base Station (BS).

ACKNOWLEDGMENTS

I would like to thank my wife and son for supporting me through many late nights. I would like to thank my parents for being my own personal cheering section. Finally, I would like to thank Rex Buddenberg, without whose insight this thesis would have never seen the light of day.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

We have come a long way since 1981 when Bill Gates was quoted as saying “No one will ever need more than 640K of RAM.” We’ve come even further since 1943, when the CEO of IBM, Thomas Watson said, “I think there is a world market for about five computers.” The fact of it is, no one could have predicted how useful or how ubiquitous computing power has become in our society. As a result, countless industries, businesses, and markets have been developed all in the name of extending more and more computing power to the consumer. Yet, stand-alone computing power can only accomplish so much. The real boon of the information age is the ability to interconnect computers and transfer information. Metcalfe’s Law states that the value of a telecommunications network is proportional to the square of the number of users of the system (n^2) [1].

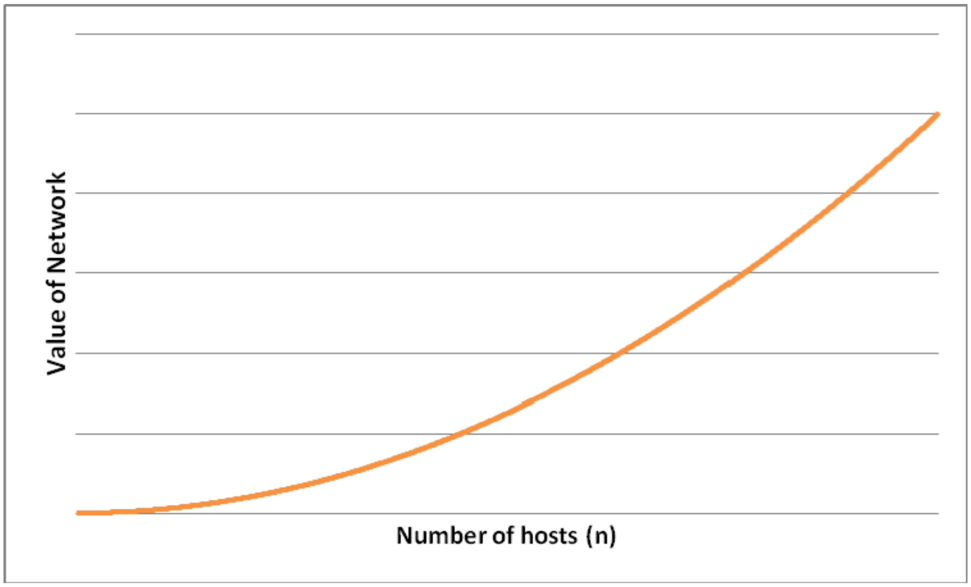

Figure 1. Metcalfe's Law

So, while users will always be added to the network at a linear rate, the potential value of the interconnections between the users goes up exponentially. Driven primarily by the end users' desire, and willingness to pay, for more capability at the edges of networking infrastructure, the trend in networking over the past 30 years has been to disperse information exchange more and more widely.

A prime example of this drive is the advent of wireless networking technology. Not content with having to move to the network, end users wanted to be able to carry the network with them. In today's market, it is almost impossible to buy a new laptop or PDA or telephone that is not network enabled. Due to this hunger for more connectivity anywhere at any time the horizon of networking technology is a world where information exchange and global connectivity on a personal level will be the norm and not the exception. One of the largest obstacles in this pursuit is the problem of the "last mile." The "last mile" refers to the distance covered by the last hop from service provider infrastructure to the end user. The high capacity links that make up the "backbone" of the service provider's network are few and have relatively large capacity; however, the links that serve the end user are significantly larger in number and simultaneously smaller in capacity.

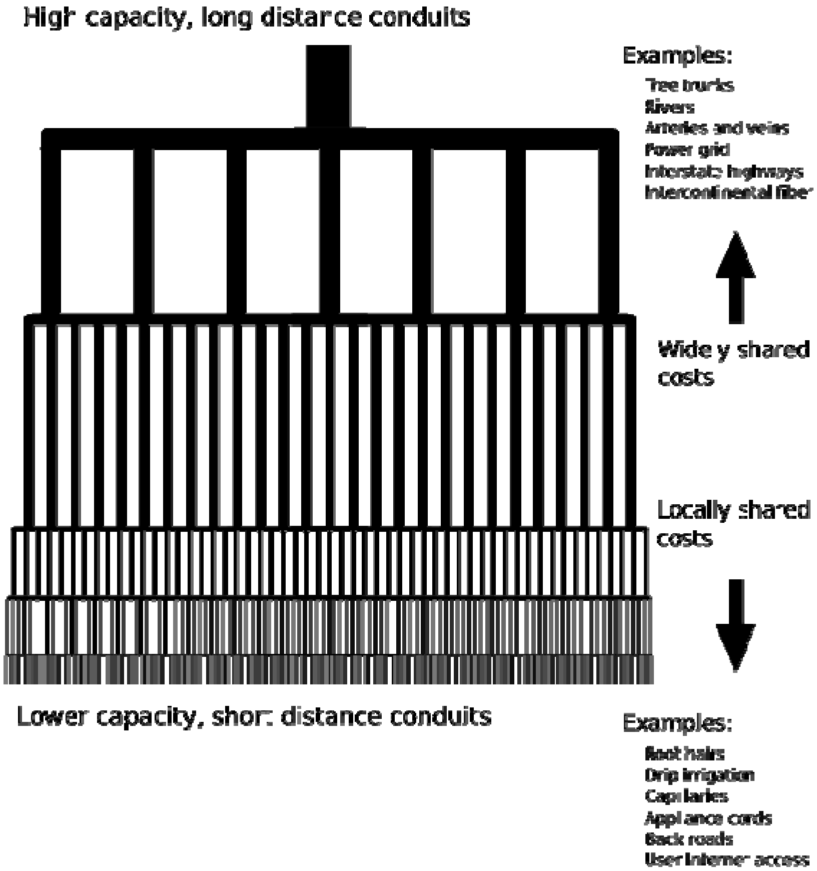

Figure 2. Illustration of "Last Mile"

Include the amount of networking overhead generated by numerous end users, in addition to the latency incurred by the many lower capacity links, and the problem of the last mile becomes obvious. Another hurdle associated with the last mile is the infrastructure required to support it. The last mile typically has a hub and spoke topology with service provider equipment acting as the hub for any number of end users. Traditionally these “spokes” are wires that are either strung overhead or buried in the ground. These wired links incur a significant amount of cost in that they are susceptible to everything from weather to back hoes, not to mention the cost associated with crossing property lines to run cable. The problems associated with getting high capacity links to numerous end users are not trivial. Fortunately, an emerging wireless networking technology is looking to solve these problems. This technology is the most promising network technology to emerge in the last ten years, geared towards satiating this hunger for ubiquitous universal connectivity and dealing with the problem of the last mile, it is IEEE Std. 802.16

The Institute of Electrical and Electronics Engineers (IEEE) is a technical professional society with over 350,000 members worldwide [7]. The IEEE is a world leader in the creation of technical standards with over 900 active industry standards. IEEE Std. 802.16 is the definition of a wireless networking technology first published by the IEEE 802.16 Working Group in 2001. Like its more common cousin, IEEE Std. 802.11 (Wi-Fi), IEEE Std. 802.16 allows end users to connect to a network over the air; however, the capabilities of IEEE Std. 802.16 are significantly different. IEEE Std. 802.16 offers an ideal point-to-point range of 30 miles (50km) with a throughput of 72 Mbps. It also offers a non-line-of-sight (NLOS) range of 4 miles and, in a point-to-multipoint distribution; the model can distribute nearly any bandwidth to almost any number of subscribers, depending on the subscriber density and network architecture [2]. IEEE Std. 802.16 is the answer to a number of problems associated with the last mile and increasing the number of entities on a network. The benefits are that it offers wireless connectivity, not in terms of feet, but in terms of miles. IEEE Std. 802.16 supports broadband speeds, and, with the publication of IEEE Std. 802.16-2005, can be used in a mobile configuration where a Subscriber Station (SS) in transit can be handed off to multiple Base Stations (BS), similar to how we use cell phones. In fact, it is likely that the next generation of cell phone networks (4G) will be based on IEEE Std. 802.16 networks. Sprint Nextel announced in 2006 its plans to rollout a nationwide 4G wireless network based on the mobile IEEE Std. 802.16 specification [3]. Since IEEE Std. 802.16 has already shown itself to be of significant interest in the commercial sector, there is no reason why the armed forces too should not look to IEEE Std. 802.16 as a viable communications method.

In that America is the sole remaining superpower, it is likely that the wars of tomorrow will no longer be fought like the wars of yesterday. No longer is the winner of a war the side that can bring the largest number of guns to bear. Today’s threats are asymmetric, today’s enemy cunning. Asymmetric warfare is a weak opponent seeking offsets against a stronger foe…Asymmetric threats will often exhibit a flagrant disregard for fighting in ways we consider either “traditional” or “fair” [4]…” While asymmetric warfare isn’t new, the capabilities and raw destructive power of modern weaponry are. In this new age, American forces not only have to fight with an agile enemy who can evanesce into the populace, but also with a system of traditional warfare that, while beneficial in previous eras, is slow moving and favors homogenous state-sponsored opposing forces. A strategy currently being employed to lessen our susceptibility to asynchronous warfare by leveraging our IT superiority is Network Centric Warfare.

Network Centric Warfare (NCW) is an information superiority-enabled concept of operations that generates increased combat power by networking sensors, decision makers, and shooters to achieve shared awareness, increased speed of command, higher tempo of operations, greater lethality, increased survivability, and a degree of self synchronization [5]. The potential gains of NCW are numerous, but all rely on the network infrastructure that supports it. In order to understand what quality attributes are important to a network supporting NCW, a hypothetical scenario might be of use.

In this scenario, a Forward Operating Base (FOB) is established to define the front lines of a battle. This FOB is near an enemy target of interest and it should be assumed close enough for the enemy to attack. Among the number of soldiers are a handful of snipers or reconnaissance soldiers who are concealed beyond the perimeter of the FOB and whose job it is to collect intelligence on enemy movement and/or stand sentry to alert the FOB of potential ambush. In this scenario it is imperative that the handful of forces that are deployed forward of the FOB perimeter remain concealed. Exposure of their position would create a life-threatening situation at the least.

In performing the functions of reconnaissance, alerting the FOB of potential attack or in the case of the sniper firing upon targets of interest, NCW enabled equipment could greatly aid the soldiers in fulfillment of their mission. Having NCW enabled equipment could enhance the soldiers’ capabilities by:

- Allowing the reconnaissance troops, with the use of a webcam, to e-mail or stream pictures of enemy encampments in real time back to the FOB, which in turn could e-mail or stream that data back to a central command.

- Allowing a central command to update in real time a list of targets of interest, enabling snipers to be constantly aware of whom to look out for.
- Allowing all forward deployed forces in the area of the FOB to have the same Common Operational Picture (COP), enhancing synchronicity of efforts, and in combination with a blue force tracker, reducing the chances of fratricide.

So what are the characteristics required of a network to support NCW? Wireless is a foregone conclusion as foot soldiers can't drag wires behind them. The network would have to be highly available. As peoples' lives may depend on the network, availability is a cornerstone quality attribute. The network would have to be secure. In general, network security can be thought of in terms of infrastructure protection and data protection; a NCW network should provide both. In that the wireless links will probably be quickly saturated, the network should be able to do some Quality of Service (QoS) to ensure the high priority traffic always gets through. All of these capabilities are available in the off-the-shelf form of 802.16. High availability is addressed in the scheduling algorithm for traffic, which ensures that even under overload conditions the network does not crash. 802.16 has several security measures to ensure the infrastructure is secure (it should be noted that data security is better treated at other levels of the network as an end-to-end solution). 802.16 also supports QoS to ensure that voice, video and data get the services they need to function properly. However, for all the gains associated with 802.16, there are still some concerns about applying the commercial product in the military environment.

IEEE Std. 802.16, as a wireless technology, transmits and receives signals through the air as its means of communication. Both ends of the connection are constantly broadcasting back and forth to negotiate session parameters, exchange security information, and pass user data. One potential downside of this constant back and forth is the risk of having the wireless signal tracked and used to compromise the position of a soldier or soldiers using IEEE Std. 802.16 technology. With very little money and information freely available on the internet, potential enemies can easily assemble a device capable of tracking the 802.16 signals emanating from equipment used by blue forces. This thesis will explore the possibility of mitigating the inherent risk of IEEE Std. 802.16 wireless transmissions by using the Simple Network Management Protocol (SNMP). SNMP is a networking protocol that can be used by network administrators to remotely configure networked devices. As SNMP can be used to remotely set configuration variables, this thesis will explore the intersection of 802.16 and SNMP to discover if there is a way to leverage the remote configuration capabilities of SNMP to allow an 802.16 end user to receive broadcast data without having to transmit in response.

A. SCOPE

This thesis will focus on the intersection of 802.16 and SNMP. Since that intersection occurs in the software aspect of 802.16, hardware issues such as wave propagation techniques, regulatory constrictions, and vendor specific hardware information are outside of the scope of this thesis. This thesis will provide an overview of both 802.16 and SNMP to provide a framework for follow-on discussion. The setup and maintenance of an 802.16 session will be analyzed in terms of the required configuration variables needed to support communication. These necessary values will then be compared with current SNMP capabilities.

B. RESEARCH OBJECTIVES

There has already been an adoption of IEEE Std. 802.16 technologies into today’s military. Historically, the model has been that U.S. military forces adopt a technology, develop it for their own uses, and then, after the bleeding edge has dulled, release it to the commercial market (airplanes, radar, and small arms). While this has worked well for military hardware, the acquisition of information systems has been another thing entirely, a good example of which would be the Joint Tactical Radio System(JTRS), a military-developed wireless radio acquisition which, to date, is a decade behind schedule and billions over budget [6]. However, with IEEE Std. 802.16, this is not the case. IEEE Std. 802.16 has been developed in its entirety with the commercial market in mind. The civilian designers never had to think about practical battlefield issues, so they did not design the 802.16 standard with any consideration that it might be used on the front lines.

To be sure, the potential benefits of using IEEE Std. 802.16 on the battlefield are innumerable; however, those benefits currently come with a price. A soldier using IEEE Std. 802.16 gear transmits wireless signals; these transmissions can be used by an adversary to locate the soldier. The objective of this thesis is to determine if there is a way to mitigate the threat of exposure and prevent loss of life.

C. RESEARCH QUESTIONS

1. What is IEEE Std. 802.16 (WiMAX)?
2. Why use IEEE Std. 802.16 in the battlefield?
3. Benefits vs. Wired/802.11/cellular technology
   (a) What improvements can we gain using 802.16 as-is?
   (b) What potential improvements can we identify which will not be addressed/ developed in and for the commercial world, and which will therefore rely on DoD investment to be realized?
4. What is the problem with using 802.16 in the battlefield?
5. What is SNMP?
6. How can SNMP be used to solve the problem?
7. By what means do 802.16 and SNMP interact?
8. Can SNMP be used to configure a silent SS connection?
9. Are there other possibilities which might accomplish the same ends?

D. ORGANIZATION

Chapter II begins by providing background information on IEEE Std. 802.16, followed by an enumeration of the various sub-components of the IEEE Std. 802.16 standard. Chapter II ends with a walkthrough of a session initiation to highlight the numerous variables required to establish and maintain an IEEE Std. 802.16 connection.

Chapter III provides background information on SNMP and will introduce the SNMP networking protocol and its relation to the 802.16 standard.

Chapter IV discusses the direct intersection of 802.16 and SNMP. In this chapter the current 802.16 capability in terms of SNMP will be explored.

Chapter V will state the conclusion of the research.

Chapter VI will discuss potential alternate solutions for accomplishing the goals stated in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. IEEE STD. 802.16

The 802.16 standard began as a project authorization request (PAR) approved by the IEEE 802 Executive Committee in 1999. With the approval of the PAR came the formation of the IEEE 802.16 working group. Contained within the larger organization of the IEEE, is the IEEE 802 LAN/MAN Standards Committee (LMSC). This group, as its name implies, is responsible for the creation of technical standards relating to Local Area Networks (LANs) and Metropolitan Area Networks (MANs). The LMSC is responsible for a number of networking technologies, the most famous of which are 802.3 Ethernet, 802.11 Wireless LAN, 802.15.1 Bluetooth, and of course 802.16 WiMAX .

The original PAR submitted for Broadband Wireless Access (BWA) established a standard for use in frequencies between 10GHz and 66GHz. This configuration, which would become known as IEEE Std. 802.16-2001, assumed a single-carrier wave, Line-of-Sight (LOS) propagation, and fixed antennas on both ends of the connection. Before the 802.16-2001 standard was even approved, the Working Group had already begun discussions involving modifications to the 2001 standard. 802.16 has seen over a dozen modifications to its standard over the past decade. Never seeming content with the status quo, the 802.16 Working Group has made a number of modifications to the standard to include operation across wider frequency ranges, NLOS environments, Mesh topologies, and mobile services. The most significant of these improvements came in the form of the IEEE Std. 802.16-2005 publication. For the first time 802.16 had a standard which addressed mobile users. By defining a "handoff" mechanism IEEE Std. 802.16 entered the market as a viable cell phone replacement technology. Looking toward the future of wireless communications in which everyone is connected everywhere all the time, the 802.16-2005 standard was the first to deliver on the promise of truly mobile Broadband services.

Today 802.16 is more useful, secure, mobile, and accessible than it has ever been. Due to the hard work of the 802.16 Working Group, 802.16 is poised to take a significant chunk out of the commercial market for Broadband access. Without the limitations of wired technologies and the relatively low cost of entry into the market for a vendor, 802.16 stands a good chance of being the method of choice to address the last mile of tomorrow. With more 802.16 standards being developed today, who can tell what IEEE Std. 802.16 will be able to offer in the future?

A. THE OSI STACK

In order to understand the inner workings of IEEE Std. 802.16, it's necessary to understand a few key concepts about networking technology. A useful model for understanding network communications comes in the form of the OSI Seven Layer model. The Seven Layer model is an abstract concept model used to assist in the creation of, and integration between, the various different transformations data must undergo when passing from application to network medium and back to application. The layering model is a fundamental tool that helps designers master the complexity of protocol software. Layering divides the complex communication problem into distinct pieces, and allows a designer to focus on one piece at a time [8]. Since the model represents a vertical hierarchy, usually drawn as boxes stacked atop one another, a generic term for any given networking protocol is a "stack."

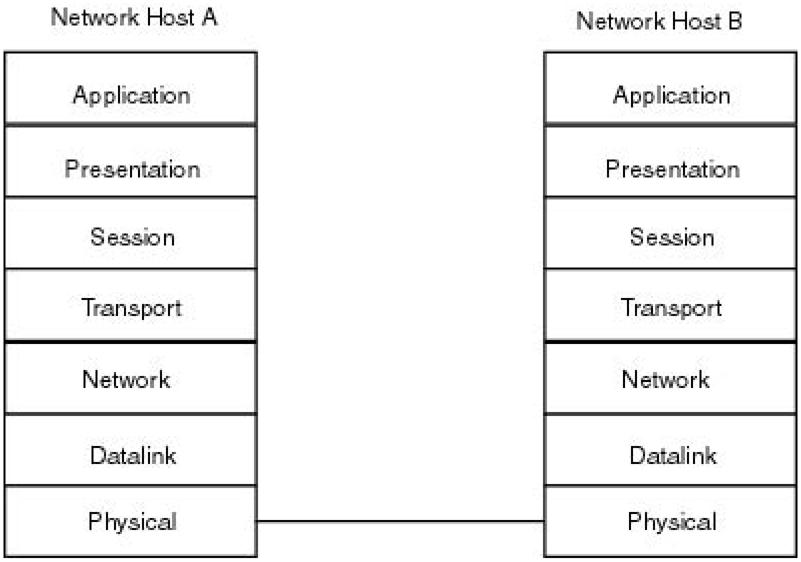

Figure 3. Figure 1. OSI 7 layer "stack"[9]

Another key concept is that of data units. When information is handed to the networking stack by an application, that information is broken down into relatively small pieces. These pieces are called Service Data Units (SDU) and Protocol Data Units (PDU), and are the basic units of information exchanged between layers of a protocol stack. SDUs are the data units that are passed from one layer to the next within the same stack; PDUs are the data units exchanged between peer layers in remote stacks. When Layer N has fully prepared a unit of data for communication to its peer layer on another host, the data unit is a PDU. Layer N then uses a service access point (SAP), which defines the interface between each layer, to pass the PDU down to Layer N-1. What Layer N-1 receives through the SAP is both the PDU of Layer N and the SDU of Layer N-1. Layer N-1 then goes about the business of transforming the SDU into a PDU of Layer N-1, at which point it uses N-1 → N-2 SAP to pass down the PDU again.

In abstract, when an application is ready to send data over a network, said application hands off the data to the Application Layer of the networking stack. The Application Layer then prepares and hands the data off to the Presentation Layer. The Presentation Layer then repeats the process and so on until the Physical Layer. At the Physical Layer, the PDU is an electrical signal that is put on a medium (copper, fiber, or air) and transferred to a recipient. When received, each data unit ascends the recipient's stack in the opposite order, this time however, the SDUs are handed upward not down. At the application layer, the SDU being handed up is the recovered application data sent from the originating host.

IEEE Std. 802.16 is a member of the IEEE 802 family of standards. Like the other 802 standards, 802.16 is built upon a layered model for communications and deals with the lowest two layers of the OSI stack, the physical, and datalink layers. The IEEE 802 reference models expound the OSI model by specifying the inner workings of each layer, defining the interfaces between layers (known as the Service Access Point or SAP), and occasionally making a few nomenclature changes. As will be shown in the following sections, while the OSI stack stops at the generic seven layers, the 802.16 standard actually defines several sublayers and the SAPs between all adjacent layers as well as the nomenclatures changes. However, it is important to note that all 802 standards perform in this way, and it is because of this modular nature that Ethernet has been able to survive for over 35 years of change in the IT and networking space. Although several different PHY approaches have been evolved through the years (Coaxial, Twisted Pair, Fiber optic), Ethernet only required that the PHY layer be redrawn and not the whole standard. As 802.16 shares these same evolutionary modular properties, it too can make small incremental changes (swapping out one PHY for another) relatively easily as the change to the PHY layer, so long as it conforms to the PHY/Datalink SAP, and it doesn't affect the rest of the networking stack.

B. 802.16 – ARCHITECTURE

Thus far we've talked about IEEE Std. 802.16 in abstract networking terms. The OSI Stack and the transmission of SDUs and PDUs could apply to almost any networking technology. The following section will focus on specific aspects of the 802.16 standard and highlight concepts that define IEEE Std. 802.16 as a unique networking technology.

In a more familiar wireless networking environment like Wi-Fi, there is no central scheduler for access to the network. In an ad hoc network each node has equal rights to the network and all nodes must contend with each other for network resources. Introduce an Access Point (AP) for connection back to the Wide Area Network (WAN), and while there is a central point of aggregation, the AP does not control per se the nodes connected to it. When introduced, an AP can send out a busy signal to help coordinate between distant transmitters, but each node is still on its own. This situation can cause additional problems in that since every node is using the same transmission power, distant transmitters can be trampled by nodes closer to the AP. In IEEE Std. 802.16 this is not the case.

In 802.16, there are two transmitting entities, the Base Station (BS) and the Subscriber Station (SS). Though both entities have instances of the 802.16 MAC layer and PHY layer, the BS and the SS perform different functions for any given connection. Chief among the differences between the BS and the SS is that the BS is responsible for maintaining many connections between many SS, while every SS has a connection with only one BS. Even connections between neighboring SS' must go through the BS. This centrally managed approach allows for scheduling of traffic, which mitigates the problem of end nodes trying to out transmit each other. Due to the greater number of concurrent connections expected of the BS it is common to find that extra resources are dedicated to it. It is also the BS which defines the characteristics of any given connection. All SS must obey the connection rules provided by the BS. This configuration has several advantages over a contention-based network in terms of scheduling, synchronization and bandwidth management.

Chief among the benefits of the BS scheduling all communications is that at all times all nodes are synchronized as to when and how to transmit and receive. This model incurs little to no overhead due to synchronization. Since the SS rarely needs to spend time contending for services, the use of bandwidth in an IEEE Std. 802.16 implementation is extremely efficient. In a Wi-Fi implementation, the introduction of additional nodes to the network results in an exponential growth in collisions brought on by two nodes trying to transmit at the same time. In an IEEE Std. 802.16 implementation this isn't the case; i.e., no matter how many nodes are added, the BS is always in control of how services are allocated. In an overload state, where a Wi-Fi network will fail due to the number of collisions, an IEEE Std. 802.16 network won't. Traffic may get "dropped on the floor" as the queues fill up, but the network itself won't cease to function.

Below is a summary of the roles and responsibilities delegated to both the BS and SS:

BS is responsible for:

- Enforcing basic MAC and PHY parameters such as frame size, ATDD, and configuration of system parameters
- Performing bandwidth allocation for DL (per connection) and UL traffic (per SS) and performing centralized QoS scheduling, based on the QoS/service parameters configured by the management system and the active bandwidth requests (BW requests) received from the SS
- Communicating the per-frame schedule to all SSs and supporting other data and management broadcast and multicast services

- Transmitting/receiving data and control information to/from one or more SSs within the same frame
- Performing connection admission control and other connection management functions
- Providing other SS support services such as ranging, clock synchronization, power control and handoff

SS is responsible for:

- Identifying the BS, acquiring PHY synchronization, obtaining MAC parameters, and joining the network if necessary
- Establishing basic connectivity, setting up additional data and management connections, and negotiation and optional parameters and needed
- Generating BW requests for connections that require such requests be generated, based on the connection profiles and traffic
- Receiving broadcast/multicast PDUs and unicast PDUs and forwarding them to the appropriate modules.
- Making local scheduling decisions based on the current demand and history of BW requests/grants, when a BS allocates bandwidth for the SS
- Transmitting only when instructed by the BS to do so or the SS has some information that qualifies for transmission in one of the slots that may cause "contention" (e.g., ranging an dBW requests in contention or broadcast allocations)
- Unless in sleep mode, receiving all schedule and channel information broadcast by the BS and obeying all medium access rules, transmitting data only when the BS allocates slots
- Performing initial ranging, maintenance ranging, power control, and other housekeeping functions [7].

As should be obvious by now, it is the BS that is connected to the infrastructure network. It is this connection to the network backbone, through the BS, that allows all of the SS to connect to the network at large. In commercial implementations of IEEE Std. 802.16, the BS is usually located on the roof of a tall building. This vantage point allows it to connect with SS either in a LOS Point to Point (PtP) configuration or in a NLOS Point to Multipoint (PmP) configuration. SS can be either an antenna placed upon the roof of another building or a card inserted into a laptop or PDA like device. One of IEEE Std. 802.16's strengths is that it supports so many different implementation combinations.

While originally created to support strictly BS to SS connectivity for fixed Broadband Wireless Access (BWA), over the years the 802.16 standard has evolved to include mobile stations and mesh topologies. Though created to use the 10-66GHz frequency range, it has grown to include lower frequency ranges to support NLOS. The 802.16 standard has also grown to include advanced antenna configurations to increase capacity and range. The 802.16 working group is constantly making changes to the 802.16 standard to make it better than before. There is no doubt that the BS and SS will continue to take on new characteristics and new implementation options.

C. 802.16 – MAC

To understand the IEEE Std. 802.16 MAC, it is helpful to expand on our existing layered model. The 802.16 standard exists at the lowest two layers of the Seven Layer model, the Datalink and Physical layers. It is worth pointing out that the model in Figure 4 is a more granular view of the lowest two layers in Figure 3. As such, layer two in the IEEE Std. 802.16 model is comprised of sub layers, respectively called the Service Specific Convergence Sublayer (CS), the Common Part Sublayer (CPS), and the Security Sublayer. Also notice that in the OSI model, layer two is called the Datalink layer and in the IEEE Std. 802.16 model it is referred to as the MAC layer, which is an example of the nomenclature changes mentioned earlier.

1. Convergence Sublayer

The IEEE Std. 802.16 MAC CS is the uppermost of the MAC sub layers. As such, it is responsible for handing SDU's back and forth between itself and the Network Layer and handing MAC Service Data Units (MSDU) back and forth between itself and the CPS. The MSDU exchange between the CS and the CPS is done by way of a service specific Service Access Point (SAP). This SAP defines the interface between two adjacent layers, and what services are available to the CS from the CPS.

The job of the Convergence Sublayer is to format the various Network layer SDU's by way of the service specific SAP in such a way that the Network Layer protocol is transparent to the CPS. This has the effect of making the CPS "protocol agnostic" [7] in addition to ensuring that, should new Network layer protocols need to be defined for IEEE Std. 802.16, only the CS needs to be redefined and not the entire MAC layer.

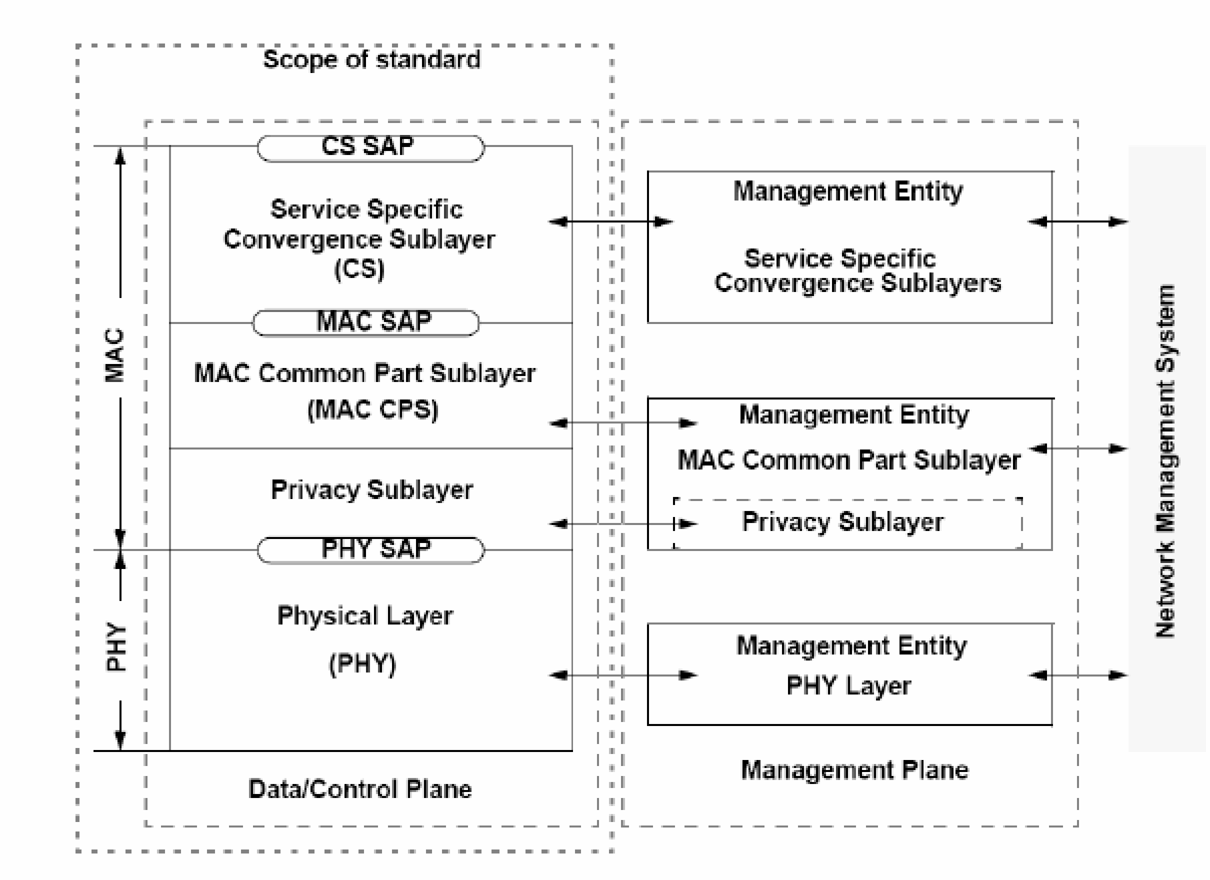

Figure 4. 802.16 Protocol Layers

Another key aspect of the IEEE Std. 802.16 CS is that it must by definition support all services that the Network layer protocol supports, meaning that there is no compromise in functionality when introducing IEEE Std. 802.16 into a network. For example, the IP protocol has IPv4 and IPv6, both of which are supported by IEEE Std.

802.16. Additionally, any Quality of Service (QoS) parameters used by a Network layer protocol must be supported by IEEE Std. 802.16

2. Common Part Sublayer

The IEEE Std. 802.16 CPS is responsible for the core MAC functions of IEEE Std. 802.16 such as medium access, connection management, and QoS functions.

a. Media Access

In a networking system there are a number of different hosts all exchanging information over a shared medium. Whether that medium is in a cable, or over the air, all the different hosts that want to use that medium cannot do so at once. The event of two hosts transmitting at the same time within the same network segment or collision domain is called a collision, the end result of which is an unusable garbled signal. In cabled networks, limiting the size of collision domains is achieved by the use of switches, which “switch” incoming traffic to the correct output port instead of broadcasting the traffic out every port. In wireless systems however, by definition, all adjacent transmitters are in the same broadcast/collision domain. Many different media access protocols have been developed over the years to deal with the problem of collisions.

The most common is Carrier Sense Multiple Access/Collision Detect (CSMA/CD) in which all stations connected to the medium listen to see if anyone else is transmitting before they attempt to transmit. If two hosts happen to transmit at the same time and a collision occurs, the hosts will “back off” for an arbitrary amount of time and then try to transmit again. While common, this type of media access scheme does not work with wireless devices. CSMA/CD works well over cable because all connected hosts can see collisions and all back off the cable. However, over an air interface, due to propagation properties such as fading and absorption, not all wireless transmitters may be aware that they are transmitting at the same time as someone else. In the case of Wi-Fi, an Access Point (AP) can raise a busy tone to let all end nodes in the area know that someone is transmitting; however, short of the busy tone as a control method, all end nodes are on their own to contend for network services alongside all other end nodes.

Any wireless service that uses this method of contention for services will always be doomed to a significant loss of efficiency due to collisions. The following sections will describe the ways in which IEEE Std. 802.16 deals with contention for the medium as well as methods for sending and receiving traffic between BS and SS.

b. Duplexing

Duplexing defines the way bi-directional traffic flows between two hosts. Any given communications device is capable of either half-duplex or full-duplex. In half-duplex, either host can send or receive, but not both at the same time. In full-duplex, both hosts can send and receive simultaneously.

c. TDD

Time Division Duplex (TDD) relies on assigning any particular host a time slot in which to transmit or receive. This method is half-duplex, as a host can either transmit or receive in any given window, but not both at the same time. In that TDD divides up the users in the time domain, it only needs to use one transmission frequency to work. IEEE Std. 802.16 also offers Adaptive Time Division Duplexing (ATDD), which can dynamically reassign time slots to reflect the pressures on the network at any given time. Using ATDD is efficient in the frequency domain as it only uses one frequency and is efficient in the time domain as it can dynamically reassign time slots where needed so no slots are wasted on idle hosts. An IEEE Std. 802.16 BS coordinates the TDD slots by the use of Upload (UL) and Download (DL) MAPS. CPS MAC Protocol Data Units (MPDU) will eventually be broadcast as PHY layer frames. TDD frames broadcast by the BS begin with both a UL and DL MAP which define the time allocation slots for all communications on the network. The DL MAP defines the DL subframe that is the part of any given frame dedicated to transmitting from the BS to an SS, just as the UL MAP defines the UL subframe. The use of subframes and slots within the subframes to transmit or receive from BS to any given SS is how IEEE Std. 802.16 manages the many connections required. In this way, IEEE Std. 802.16 gains great efficiencies of the available bandwidth as every slot can be dedicated for transmission and no two hosts will try to transmit at the same time.

d. FDD

Frequency Division Duplex (FDD) uses two separate frequencies to transmit and receive data. This use of two different frequencies means that any given host can operate in full-duplex mode by transmitting on one frequency and receiving on the other. FDD is very good at dealing with symmetric traffic, as the two frequency bands used are of equal bandwidth. However, significant efficiency can be lost if FDD is used for asymmetric traffic where only one host is transmitting. This imbalance effectively wastes the unused UL frequency band. FDD in IEEE Std. 802.16 also uses frames, and these frames also use DL and UL subframes. However, the major difference between TDD and FDD is that the UL and DL frames in FDD can overlap in time. In TDD an end user gets all of the bandwidth some of the time, and in FDD and end user gets some of the bandwidth all of the time.

3. Multiple Access

a. TDMA and FDMA

TDMA and FDMA are to Multiple Access what TDD and FDD are to Duplexing. TDMA shares the communication medium by allow a particular host to transmit or receive only in their allotted time slot. This method of media access is best used with a central controller that assigns the time slots for each communicating host. As such, IEEE Std. 802.16 is a perfect fit for TDMA, as its BS can be the central controller to dynamically assign, in the case of ATDD, user time slots. IEEE Std. 802.16 gains other efficiencies in using TDMA such as almost no contention on the network. Through the use of the UL and DL MAPS, IEEE Std. 802.16 is able to ensure that no connected hosts will cause collisions by transmitting at the same time. Even in the event that a new user wants to join the network, there are time slots in the UL subframe dedicated to network entry. This means that any new host wanting to join the network needs only to listen for an UL MAP to determine when to send an admission request to the BS. In this way multiple hosts can use the media with little to no overhead.

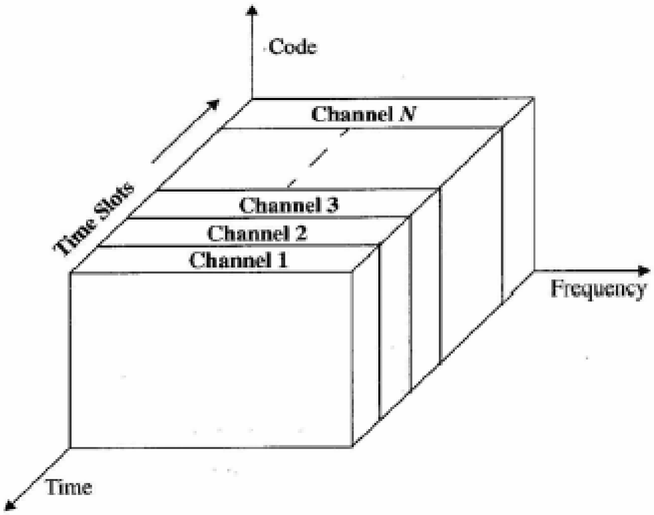

Figure 5. Model of TDMA

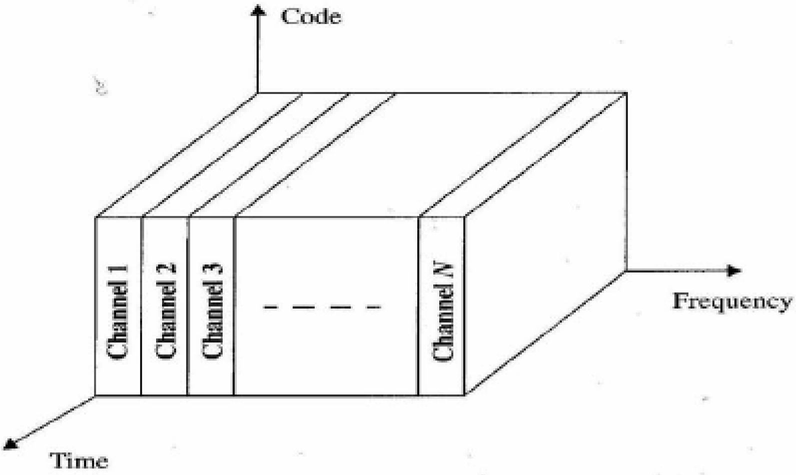

Figure 6. Model of FDMA

FDMA uses a separate frequency for each host allowing all hosts access to the medium at the same time. However, each host must have a channel sufficiently distant from other channels so as not to cause interference. Spectral efficiency must always be kept in mind when implementing FDMA, because while TDMA can dynamically assign time slots as network needs change, FDMA cannot. FDMA also has a collision avoidance scheme in that no two end users will use the same frequency to transmit/receive, and a frequency range is set aside for new users wanting to connect to the network. It is important to note that both TDMA and FDMA can use either TDD or FDD.

b. OFDMA

Orthogonal Frequency Division Multiplexing (OFDM) is a multiplexing technique that subdivides the available bandwidth into multiple orthogonal frequency sub-carriers [10]. OFDMA is a multiple access scheme based on OFDM. Whereas OFDM multiplexes a single data stream across multiple sub-channels, OFDMA assigns an end user a sub-channel (or multiple sub-channels) and transmits multiple user streams at the same time over one frequency. One of the main benefits of this method is that it can be controlled to allow any one user at any given time to have more of the sub channels than other users. In this way, OFDMA can support QoS by allowing delay intolerant traffic to take up more of the spectrum.

4. Connection Management

In IEEE Std. 802.16 the CPS is responsible for keeping track of and supporting the various over-the-air connections. There are a number of performance aspects of wireless communication that require constant adjustment and tuning for reliability. Here are a few:

- Initial Ranging and Sign On – When a user wishes to join the network, after listening for the appropriate time or frequency to transmit, there are a number of configuration values that must be negotiated between the BS and SS. Variables such as power levels, authentication, registration, frame size, all need to be communicated between BS and SS before the SS can join the network.
- Bandwidth Requests – certain end users on the system may require more bandwidth to support traffic-intense connections. The CPS needs to keep track of bandwidth assigned to specific users and from time to time, increase the allotted amount of bandwidth. In TDMA this means assigning more time slots; in CDMA

this means more of the shared noise space, and in OFDMA this means allocating more of the sub-carrier frequencies.

- Automatic Repeat Requests (ARQ) – The ARQ is a class of retransmission algorithms for supporting reliable delivery in the presence of errors [6]. ARQs are used when an end system notices missing or corrupt MPDUs in session traffic, and are a means to recover the original MPDUs so as not to degrade the session. As ARQ is by definition a two-way communication, it is outside the scope of this thesis.
- CID – Most networking technologies use the 48bit MAC addresses burned into the Network Interface Card (NIC) to uniquely identify neighboring hosts. IEEE Std. 802.16 uses a 16bit CID to identify information exchanged between the BS and SS. This more compact form of identification is used in several ways to support and track information exchange between the BS and SS. While the application of each of the following CIDs is beyond the scope of this thesis, it is important to recognize that the CPS is responsible for the management of all the various types of CIDs and that each serves a unique purpose over the course of a IEEE Std. 802.16 connection:
    - Initial Ranging CID
    - Primary Management CID
    - Secondary Management CID
    - Transport CID
    - AAS Initial Ranging CID
    - Multicast Polling CID
    - Padding CID
- SFID – Service flows are used by the IEEE Std. 802.16 MAC to efficiently support per-connection services such as QoS [7]. When an IEEE Std. 802.16 SS is assigned its Basic CID, SFIDs can be mapped to that CID to support various service flows. These flows are dynamic and can be modified at any time during the session on a per-frame/per-SS basis.

a. QoS

QoS, at its base, is recognition that different types of network traffic (voice, video, data) have different resource needs. For example, data transmission is usually fairly delay tolerant. Provided that there is adequate buffer space in the receiving host, data packets can arrive out of order or inconsistently and through various best effort techniques. Even if a packet is lost it can be accounted for and retransmitted. Voice or video packets on the other hand cannot afford to arrive out of order or be retransmitted. QoS deals with the differences between network traffic types to help ensure that no matter what information is being passed over a link, it will be received in the way that it needs to be received. QoS can commonly be measured as a function of:

> ***Throughput:*** Throughput is typically specified in bits per second, although it can be also specified in bytes per second or packets per second, depending on the application. The actual application throughput depends on a variety of factors, including packet size, overhead and retransmissions. The achievable throughput is bound by the maximum and instantaneous capacity of the network, also specified in bits per second.
>
> ***Delay:*** The delay, also known as *latency*, is the time taken for the information to travel from a source to a destination and vice versa. The delay is specified in nits of time, such as milliseconds.
>
> ***Jitter:*** Jitter is the variation in delay. It is very important to many interactive applications such as voice and video.
>
> ***Loss:*** The packet loss in any system is typically defined as a percentage. Data applications can tolerate some delay and jitter, but they cannot readily tolerate packet loss. If a data packet is lost, it must be retransmitted by some layer (e.g., network or application) for the application to work correctly. On the other hand, real-time applications such as voice and video can tolerate a small percentage of loss. Re- can be used to compensate for packet loss at the cost of increased latency and reduced available capacity [7].

IEEE Std. 802.16 has a number of ways to help guarantee a certain level of service for specific service flows. QoS starts for IEEE Std. 802.16 in the CS. As packets are handed down to the service specific CS, the CS is responsible for supporting all services of the higher protocol. That means that, if the layer three protocol supports a

QoS mechanism, the appropriate CS must also support the same QoS mechanisms as it hands the packet down to the CS. At this point the CPS can use a number of different techniques to efficiently order packets for transmission such as Admission Control, Traffic Classification, and Scheduling. The SFIDs are used to track the various flows and assign resources to those identified and needing them.

Bandwidth Requests are another method available to IEEE Std. 802.16 to support QoS. As mentioned earlier, it is the responsibility of the SS to request more bandwidth when needed and the responsibility of the BS to assign network resources as they become available. Using bandwidth requests an SS can help guarantee that its traffic is getting the resources needed. In combination with the QoS capabilities of the BS, both ends of the connection can take an active part in supporting QoS.

5. Security Sublayer

The Security Sublayer was designed to secure the connection over the IEEE Std. 802.16 link. As the security methods are applied at the sending BS/SS and then stripped off at the receiving BS/SS, the security measures employed in an IEEE Std. 802.16 connection do not persist with the network traffic outside the BS/SS to BS/SS link. The primary focus of the Security Sublayer is to prevent Theft of Service (TOS) and, to an extent, Denial of Service (DOS); the secondary effect is to protect the data.

The Security Sublayer of IEEE Std. 802.16 comes in two different parts. The first is called Privacy Key Management (PKM), which, as its name implies, deals with secure key exchange between BS and SS. The PKM uses X.509 digital certificates, the RSA public key encryption algorithm, and strong encryption algorithms to perform key exchanges between SS and BS [11]. PKM is how IEEE Std. 802.16 goes about the business of distributing keys between BS and SS to support traffic encryption. PKM uses Public Key Cryptography (PKC) to establish the set of keys it will use to encrypt payload traffic. PKC relies on two different keys — the public key and the private key. The keys are related in such a way that traffic encrypted by one can only be decrypted by the other. Each SS has a unique vendor supplied X.509 certificate. This certificate contains the PKC keys that will be used to authenticate the SS to the BS. When a new host sends its initial request to join the network there are no shared keys between the SS and BS. To establish encryption keys the SS must send the BS its X.509 certificate that contains a copy of the SS' public key. If the certificate checks out, the BS will then use the SS public key to encrypt a shared key to be used for future authentication and key generation. When the SS receives this new encrypted packet it uses its private key to decrypt the message, which contains the Authentication Key (AK) between the BS and the SS. Now that the BS and the SS have a shared AK, they can use that AK as the basis for creating more shared keys. Neither IEEE Std. 802.16-2001 nor IEEE Std. 802.16-2004 specify a means for the SS to authenticate a BS, creating a situation where a malicious BS could be inserted into the network; however, IEEE Std. 802.16-2005 remedies this situation by providing a mechanism for BS authentication. By this means of key exchanges the BS can ensure that no unauthorized SS gains access to network resources. The key exchange in the PKM process is also the base for the traffic encryption method in the Security Sublayer.

The second part of the Security Sublayer uses a frame encapsulation protocol, which defines techniques for encrypting MAC payload data. IEEE Std. 802.16 can use either Data Encryption Standard (DES) or Advanced Encryption Standard (AES) to encrypt its payload data. DES is a cipher that was adopted by the United States as a federal standard in 1976. It uses a 56bit key to encrypt data that can then only be decrypted with the same key. Unfortunately, given Moore's Law, DES' 56bit key is no longer a computational barrier; thus, the encryption offered by it is considered weak. AES is another encryption standard adopted by the United States. AES was adopted in 2001 and has variable key sizes of 128, 192, and 256 bits, though IEEE Std. 802.16 only uses the 128 and 256bit key lengths. AES is considered the current standard for secure encryption. As they apply to IEEE Std. 802.16, DES does not provide data authenticity, or strong encryption; however, DES does not increase the size of the MPDU payload causing no additional overhead by its use. AES, on the other hand, does provide strong authenticity, data integrity, and confidentiality while adding 12bytes to each and every MPDU transmitted causing a significant overhead increase. Both DES and AES require a key for encryption and decryption. These keys used for frame encryption are all created by way of the AK established in the PKM process. There are, however, some noteworthy issues associated with the frame encryption concept.

First, as mentioned above, encrypting the payload of a frame only supplies security over the relatively short hop of the local LAN. If traffic needs to traverse a number of different networks to get from source to destination, any security at Layer 2 of the OSI model is only good for one specific Layer 2 link. As no sender knows exactly what path their network traffic will take, trusting in Layer 2 security is a dangerous gamble as there is no guarantee that every link along the way will be secured. Furthermore, if one implements security at a higher layer of the OSI stack (for example, a Layer 3 Virtual Private Network (VPN)), then the Layer 2 security measure is moot, as the traffic is already encrypted when it enters the Layer 2 link. It is for these reasons that the frame encryption capabilities of the Security Sublayer are ancillary to the TOS/DOS protections granted by the PKM process.

As mentioned, the AK is created during the initial ranging period of a BS/SS connection. The AK is used as the base for all other unique keys between the BS and each individual SS respectively. Security Associations (SA) are a set of security related variables that must be negotiated between the SS and BS before they can encrypt traffic. When an SS sends its initial request and X.509 certificate to the BS, it also sends along a list of all of the SAs that the SS is capable of using. The BS is then given the list of possible SAs, chooses one and relates that choice back to the SS along with a Traffic Encryption Key (TEK). All connections between a BS and SS must have at least one SA, and all SAs must have at least one TEK. The TEK is in charge of the keying material used for any particular SA. For example if SA_1 defines DES as the encryption standard to be used, the TEK for SA_1 holds the 56-bit encryption key used to encrypt the traffic sent over that SA. Although each BS/SS pairing has one unique SA, additional static and dynamic SAs can be created for new data streams on a one-to-one or one-to-many basis. Each MPDU is associated with an SA, and before it is transmitted, undergoes the permutations dictated by that particular SA and encrypted using the keys stored in the TEK for that SA. Once the encrypted data is received, the receiving station uses the CID to select the appropriate SA for decrypting the data and recovers the original payload.

While the security processes and procedures used in IEEE Std. 802.16 are useful and robust, they ultimately come up short, as any security measure at the lower layers of the OSI model must. A time-tested method of securing information has been to encrypt it such that only the sender and receiver can decrypt it. While IEEE Std. 802.16 does use encryption, it can only encrypt the traffic that passes over its interfaces. Unfortunately, encryption based security at Layers 1, 2 and 3 of the OSI model are not truly end-to-end security solutions. For example, in a hypothetical IEEE Std. 802.16 implementation, the IEEE Std. 802.16 security is only in play for the brief time that the traffic is sent from BS to SS. Before the traffic gets to the BS, and after the traffic is received at the SS, IEEE Std. 802.16 cannot protect the data. It is noteworthy that IEEE Std. 802.16 security is more about protecting the infrastructure and avoiding theft of service, than it is about protecting the data passing over the network. Data protection should start at the highest level of the OSI model, which would allow for true end-to-end data protection.

D. PHY LAYER

While IEEE Std. 802.16 has only one MAC Layer, there are several PHY layers defined in the standard. The first of the PHYs published in the original standard for IEEE Std. 802.16 is known as WirelessMAN-SC and is a single carrier, point-to-point PHY for operation in the 10-66GHz frequency range. Due to the propagation characteristics of this frequency range, Line of Sight is an operational requirement. Hence, one of the largest problems associated with WirelessMAN-SC is signal degradation brought on by inclement weather. Another limitation of WirelessMAN-SC is that both the BS and the SS must remain fixed. The characteristics make WirelessMAN-SC a good option for infrastructure links (replacing or obviating the need for some cabled infrastructure links), but unfeasible as a solution as a cell phone replacement technology.

Toward this end, the IEEE Std. 802.16 working group has, over the past decade, created a number of other PHYs to deal with these issues. The second PHY published is WirelessMAN-SCa. This second PHY was created to work in the 2-11GHz frequency range, meaning that it could operate in a NLOS fashion. Additionally, SCa included provisions for additional modulation techniques such as BPSK and 256-QAM, as well as support for beam forming which with advanced antennas, could be used to focus and extend the range of SCa's wireless signal. So far, the PHYs available for IEEE Std. 802.16 were single carrier, and while SCa addressed the problem of BWA by moving to the 2-11GHz frequency range, the single carrier scheme still had difficulties in a multipath environment.

Multipath is a reality of wireless transmissions in that when a signal is sent out over the air it reflects off of physical objects in the environment. Not only does the receiver receive the intended signal, but also the dozens of copies reflected off of objects in the environment. These copies tend to interfere with the original signal, a phenomenon known as Intersymbol Interference (ISI), by their energy either constructively or destructively interfering with the intended signal. The next development in the PHY addresses ISI.

The WirelessMAN-OFDM PHY is based on Orthogonal Frequency Division Multiplex (OFDM). OFDM is based on a mathematical process called Fast Fourier Transform (FFT), which enables 52 channels to overlap without losing their individual characteristics (orthogonality) [9]. WirelessMAN-OFDM deals with the problem of multipath in several ways. First the OFDM addresses all of the subchannels to the intended receiver. As only a portion of the subchannels will degrade due to multipath, the relative signal degradation is limited. Furthermore, WirelessMAN-OFDM uses a relatively large guard band between symbols to help the receiver distinguish between received symbols. OFDM can use larger guard bands than a single carrier system because it is able to pack so much more information in a single symbol using the many subchannels.

The next PHY standard released for IEEE Std. 802.16 is the Wireless-MAN OFDMA PHY. Like OFDM PHY, the OFDMA PHY uses OFDM to multiplex numerous subchannels to be sent all at the same time, however while the OFDM PHY uses TDMA time slotting for multiple access, the OFDMA PHY uses the various subchannels to distinguish multiple SS'. The OFDMA PHY is significantly greater than the OFDM PHY, both in terms of flexibility and in terms of complexity. As discussed earlier, the OFDMA PHY can be extremely useful in its capability to reassign subchannels on an as-needed basis to support QoS. For this reason, the OFDMA PHY is getting the most attention in conjunction with the IEEE Std. 802.16-2005 standard for mobile systems, as they both represent the future IEEE Std. 802.16 in terms of getting the most amount of capability in the most amount of end users hands. Since each new iteration of the IEEE Std. 802.16 PHY operates in the multipath resistant frequency range of 2-11GHz and the latest PHYs deal specifically with working in an environment rife with buildings and other obstructions, it's clear that NLOS operation is driving the development of future IEEE Std. 802.16 PHYs. This development trend is in line with the desire to make IEEE Std. 802.16 not just another, but the best option for the future of residential BWA.

One additional note: as stated, one of the benefits of a layered model is isolation, meaning that you can modify or improve one layer of the model without having to redefine the entire structure. As is evident, IEEE Std. 802.16 makes great use of this capability, by constantly creating new PHYs to work in new environments. So long as the rules of the MAC/PHY SAP are observed, new PHYs can be defined arbitrarily. As creating a new PHY is relatively trivial, it should be noted that IEEE Std. 802.16 has little in the way of limitations in terms of wireless transmission methods it can ultimately support.

1. Modulation

One other PHY aspect of an IEEE Std. 802.16 connection is the modulation scheme used to send data over the air. Modulation speaks to the way signal states are manipulated to convey information. Modulation is nothing new and more familiar than one might think at first. Frequency Modulation (FM) and Amplitude Modulation (AM) have been used in radio communications for the better part of the last century. In the case of IEEE Std. 802.16, modulation is the same thing in that it uses state changes in a radio signal to convey information. While IEEE Std. 802.16 could use FM and AM, these analog methods of modulation are inefficient by today's standards because they only communicate one bit of data per state change (symbol). The modulation schemes available to IEEE Std. 802.16 are 64-QAM, 16-QAM, and QPSK; all of which use combinations of modulations to impart any given signal with more than one bit. For example, using a sine wave, we can communicate one bit of information by changing either the amplitude or the frequency. However, if per symbol, we change the amplitude and the frequency together, we now have the capability to communicate two bits per symbol. If we change the amplitude, frequency, and shift the phase, we can now communicate three bits per symbol, and so on.

The various possible bitwise combinations useable by these modulation schemes can be represented in a constellation. In the constellation for 16-QAM, each symbol is capable of conveying one of sixteen four-bit signals, for QPSK, one of four two-bit signals. In using the various modulation methods a tradeoff must be recognized. While the QPSK constellation can only carry two bits per symbol, the signals are relatively spread out and easy to distinguish from each other. In the 16-QAM example, significantly more information per symbol can be sent, but the signals being used are much closer together. In less than ideal transmitting conditions the closeness of these signal values can be problematic.

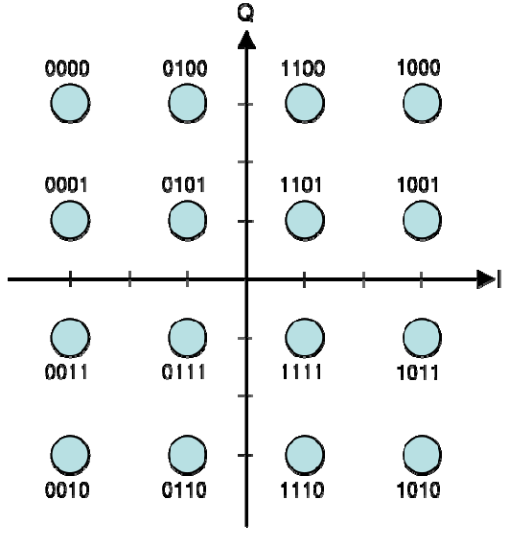

Figure 7. 16-QAM Constellation

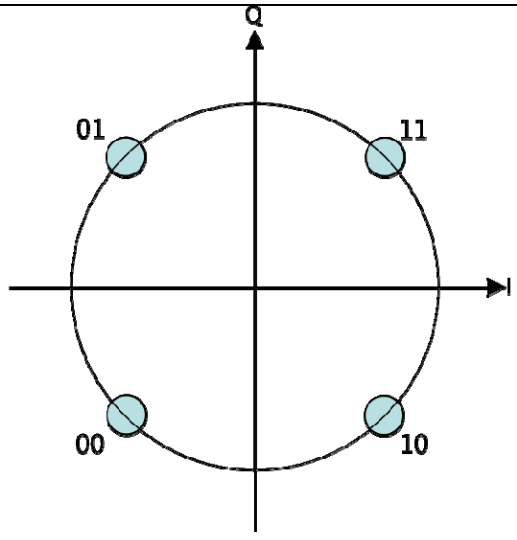

Figure 8. QPSK Constellation

In a wired networking environment, steps can be taken to ensure that the transmission media is in top shape, and free from interference at all times. Indeed, modern copper and fiber cabling come with a number of innovations to ensure the quality of the signals passing over them at any given time. Wireless communications however aren’t so fortunate; they are at the mercy of the environment. Weather, obstacles, and other wireless transmissions all affect the potential quality of any single link. These factors, in combination with natural propagation aspects such as attenuation and reflection, combine to add noise to the received signal. This noise can tend to shift the received signal so that it isn’t squarely over one of the available bit values. IEEE Std. 802.16 deals with this by dynamically choosing the right modulation scheme for the current transmission conditions, ensuring that the best connection available in terms of Signal-to-Noise Ratio (SNR) is the one being used. In poor transmission conditions or over very long distances, the BS will choose the QPSK modulation method as it is the most robust against the effects of noise. However, if the BS and SS are close and the environment is conducive to transmission, then the BS will choose the 64-QAM modulation, which allows for 8 bits per symbol. Again, the BS can do this dynamically on a per SS basis, allowing for a constant system-wide best fit balance between range and bandwidth.

E. CONNECTION SETUP

Now that we have established some basic concepts about the inner workings of IEEE Std. 802.16, let’s take a look at how a connection is established to see all the parts at work. The first thing an SS does upon powering on is refer to a list of available frequencies for use provided by the vendor. The SS will then listen on these frequencies to see if there is a BS within range. Note: it is assumed that there is already a BS up and running because if there were not, the rest of the exercise would be moot. Provided that a BS is in range, the BS will be transmitting the DCD and UCD messages at regular intervals. The DL Channel Descriptor (DCD) and UL Channel Descriptor (UCD) are specialized messages that describe the nature of the PHY scheme being used. The DCD and UCD contain information such as PHY modulation, frame duration, and Forward

Error Correction (FEC) parameters. FEC is a PHY level method of correcting errors in a transmission by adding redundancy to the transmitted signal. DCD and UCD messages are important as each 800.16 PHY goes about initial ranging differently. Another key piece of information needed by the SS are the frame preambles, which contain the DL and UL MAPs. The SS will use the UL MAP to locate the initial ranging time slot or frequency advertised by the BS, in addition to the backoff method suggested by the BS if two SS choose the same initial ranging slot.

Once the SS has this information and knows how to go about forming a communication with the BS, the next step is to begin the initial ranging process. The initial ranging process begins with the SS sending a RNG-REQ to the BS. The SS starts by using the lowest power setting available so as to not interfere with neighboring SS, and increases power incrementally as it continues to send RNG-REQ. This will continue until the SS receives a RNG-RSP from the nearest BS. When a BS successfully receives a RNG-REQ, the RNG-RSP it sends back to the SS will contain power, frequency, and timing adjustments for continued connection to the BS, along with the Basic CID, the 16-bit number that will associate this SS with the BS.

It is worth noting that, while the SS using the lowest possible power setting was originally devised as a means of preserving battery life, it has a beneficial side effect aligned with the goal of this thesis. By IEEE Std. 802.16 SS using the lowest power setting, unlike Wi-Fi in which all end nodes transmit at the same power level, the SS make themselves harder to detect. Effectively, to detect an SS using this output algorithm, a detector would need to be no further away from the SS than the BS. Once the initial power levels have been sent, the SS will continue to use the lowest power setting until a change in the environment requires a more powerful signal to communicate with the BS, at which point the new lowest power signal will be chosen. While clobbering other transmitters is not so much a concern due to the scheduling algorithm, this power saving method provides good transmission security as a bonus. The next step of session initiation is the basic capability negotiation.

The SS only comes with a certain number of capabilities in terms of maximum power output, modulation schemes, etc. The basic capability negotiation is where the SS communicates to the BS exactly what it is capable of, using a SBC-REQ message. This message, when received by the BS, contains a list of all available communication options that the SS can support. The BS then generates a SBC-RSP message detailing which of the SS capability options will be used for future communication. At this point, the BS and the SS can communicate effectively; however the BS, though capable of communication with the SS, does not know if the SS is allowed to communicate on the network.

The next step in the process is the authentication phase. The authentication phase is managed by a number of PKM-REQ and PKM-RSP messages from SS to BS, and vice versa. First, the SS sends two X.509 certificates to the BS, along with a list of security services that the SS is capable of supporting. The two X.509 certificates are a manufacturer-signed certificate for the SS and the manufacturer's certificate. Using the manufacturer's certificate (especially if SS and BS share manufacturers), the BS can then validate the SS' X.509 certificate validating it as a legitimate network host. The BS then uses the SS public key supplied by the X.509 certificate to encrypt the AK, which will be used as the basis for creating all future encryption keys between the BS and SS. Next, the BS packages the AK, AK lifetime value, and list of SAs the SS can use and sends that back as a PKM-RSP.

Now that the SS has properly authenticated with the BS, the SS needs keys to support the various SAs it will be using to encrypt traffic. The SS sends a request for encryption keys signed with its digital certificate. As the BS has already authenticated the SS, it uses the AK to create additional keys (TEKs) and sends them back to the SS. Now the SS begins the registration process in which the SS and BS negotiate high layer issues like IP version, and various QoS methods supported. Once the registration phase is complete, the SS can request an IP address via DHCP, download a configuration file from a TFTP server (if managed), and get network time using the Network Time Protocol. Once complete, before any user data can be sent from the SS, the BS sends a Dynamic Service Addition Request (DSA-REQ) to the SS. As service flows are primarily assigned by the BS, not the SS, the DSA-REQ established the first service flows available to a SS. Once the DSA handshake is complete, the SS is a fully functional member of the network and can send and receive end user data.

One stark contrast between IEEE Std. 802.11 and IEEE Std. 802.16 is that IEEE Std. 802.11 is contention based and IEEE Std. 802.16 is not. The initial period — where the SS has received the UL MAP and knows the timing for its initial RNG-REQ — is the only time that the SS will have to contend for services from the BS. It is possible that two SS will choose the same time slot to send their RNG-REQ; however, unlike a node in an 802.11 network, this will be the last time that the SS must contend with other nearby SS for BS services. After the initial RNG-REQ has been received by the BS, a new allocation is provided to the SS and it falls neatly in line with all other nearby SS. This fraction of time dedicated to contention in IEEE Std. 802.16 is one of the reasons it is so much more stable under heavy network loads.

It is worth noting that the process for initialization was designed from the ground up to require as little end user participation as possible. Both the BS and the SS have their specific roles and, between the two, the network sign on procedure happens ideally with no user intervention whatsoever. Also worthy of note is that SS can come preconfigured to work with a specific BS. This has the advantage of preemptively solving the problem of SS getting confused by too many closely spaced BS. Additionally, this also has the advantage of solving the malicious BS problem created by authenticating the SS to BS, but not the BS to any SS (though this problem was also addressed in IEEE Std. 802.16-2005).

III. SNMP

A. INTRODUCTION

Since the early days of the Internet, the number of end users has grown rapidly and shows no sign of stopping. As the number of users has grown, so has the number of infrastructure devices supporting them. In the early days of the internet, as in any small network, it was feasible for a network administrator, by means of constantly logging into and reviewing log data, to manage the networking devices. However, this method can only be considered useful in the smallest of networks. When networks grow to thousands of hosts and hundreds of infrastructure components (routers, switches, hubs, firewalls, etc.), a network management scheme becomes necessary.

B. NETWORK MANAGEMENT

Network management has been studied extensively over the last twenty years. One beneficial outcome of that continued study and refinement is the subdivision of Network Management into five logical groupings. The groupings are Fault Management, Configuration Management, Accounting Management, Performance Management, and Security Management, known together as FCAPS. FCAPS today is the most common framework to address the minimum standard of considerations when implementing a network management scheme.

Fault Management – "fault management detects, locates, and corrects problem in the network hardware and software [11]." FM makes a distinction between errors and faults in that a number of errors are expected in any given network, however any severe event or the accumulation of a number of errors can be considered a fault. In order for FM to work, all of the managed devices must have an established baseline of operation. It is the deviation from this established baseline that can trigger a fault event, which then initiates other related processes to correct the recorded fault. While the ultimate goal is to be one step ahead of faults (a point we will soon come to), FM deals with the efficient, timely detection and remediation of a fault.

Configuration Management – Configuration Management (CM) deals with not only the individual configurations of managed devices on a network, but also the configuration of the network itself. Good CM keeps a current record of the configuration of all managed devices. Furthermore CM records what changes are made, when the changes are made, and by whom. This accounting for movement on the network can be invaluable during an unexpected failure. With proper CM in place, it is a trivial matter to review all of the recent changes on the network to see if the problem was caused by a rogue configuration entry. CM is also concerned with the topology of the network. The topology of the network is a representation of how all the varied devices interconnect with one another. In a network this information is paramount, as the flow of data from device to device can be seriously impacted by a network path being moved from one location to another.

Accounting Management – Accounting Management deals with tracking end users and recording their resource usage. Primarily the interest of billing departments in a business sense, in networking, AM measures the volume of traffic a user is exchanging over the network, or the amount of time a user is connected to the network, and whether that user is causing an undue burden to the network. This information can then be turned into a billing summary to charge an end user appropriately for usage of networking resources, or be used as justification for removing the end user from the network. Additionally, AM can be used for forecasting network demands. Using the metrics collected, a network manager can see the gradual climb in network usage over time, which he or she can then present to upper management as a justification for more equipment etc.

Performance Management – Whereas, as mentioned earlier, FM answers the question of what to do when a fault occurs, Performance Management (PM) is concerned with how well the network is functioning before a fault occurs. PM allows a network manager to be proactive in fault detection instead of reactive. PM is also a very good tool to use for forecasting. Instead of looking at traffic on a per user basis, PM is concerned with the overall traffic patterns over the entire network and can be used to chart changes over time allowing a network manager to anticipate future requirements. PM uses metrics like throughput (amount of information exchanged over time) and latency (measure of elapsed time from request to expected response) to measure the health of a network and the devices in it. Using these metrics a network manager can "tune" his or her network to get the most out of it or align networking resources to high traffic areas.

Security Management – Security Management deals with access to various systems. SM involves the process of keeping track of who has access to what and ensuring that users can only get to the resources they are allowed to use. Least Privilege is a useful security related concept that states that a user should have just so much access such that they can perform their required tasks, and no more. SM is a useful method to help enforce Least Privilege and to keep the network secure from both inside and outside threats.

C. SNMP HISTORY

Though SNMP was originally created to deal with Fault Management and Configuration Management, over the years it has grown to include the three remaining aspects of the model. Shortly we will see how SNMP aligns with the FCAPS model; however, as the discussion of SNMP thus far has been generic, several specific concepts must be examined before we can continue.

In 1988, we saw the first implementation of Simple Network Management Protocol (SNMP), a networking protocol geared towards the end of creating a simple management method where administrators could efficiently manage a large number of remote assets. While SNMP was originally created as a stop gap measure until a better management protocol was created it has over time become the de facto standard for network management around the world [12].

SNMP was originally conceived to manage the bridges, routers and gateways that made up the backbone of the early internet. As an evolution of Simple Gateway Management Protocol (SGMP), it was designed to exchange rudimentary operating statistics of early networking devices and provide a format to transmit management messages. Today, everything from routers to printers to internet-enabled refrigerators can be managed by this lightweight open-standard protocol.

SNMP is capable of enabling and disabling ports on a router or switch, it can tell a manager how many bytes have passed through a particular interface, and it can even record and monitor the operating temperature of particular assets. Furthermore, SNMP is not only capable of managing hardware assets; increasingly software applications (including Operating Systems) are being produced with SNMP capability built in by manufacturers.

SNMP is published by the Internet Engineering Task Force (ITEM), an open standards body that creates internetworking standards. The IETF uses Request for Comments (RFC) as a means of submittal and tracking bodies of work on their way to the "standard" status. RFCs number assignments are permanent; as such, older RFCs are superseded by newer RFCs and given the status of "historical." The first version of SNMP, SNMPv1, was established by RFC 1157 and is now considered historical. SNMPv2 was first implemented in 1996 and is considered historical as well. The only current SNMP standard is SNMPv3, which was first introduced in 2002.

Each new iteration of SNMP builds off of the foundations of the previous version. SNMPv1 was a very simple scheme for recording and collecting management data from network assets. As such, the number of distinct data elements turned out to be lacking and, while there were efforts to include security into the model, the working group could not decide on a singular solution. Consequently, the working group pushed SNMPv1 out the door with a weak security implementation, believing it better to have a security deficient version of SNMP than no version at all. SNMPv1 authenticated messages from manager to managed asset by using a "community string" which is little more than a password known to both the manager and the managed asset. The critical flaw in SNMPv1 security is that when messages were sent between SNMP entities they were sent "in the clear," meaning that a community string was put on the wire exactly as it was typed, without encryption. This allowed for anyone connected to the network to "sniff" the SNMP traffic and easily learn the community names used. Keeping in mind that anyone armed with the appropriate community string could tell the networking equipment to shut off ports or redirect traffic, an unauthorized user could cause complete havoc. This limitation inherent in SNMPv1 persisted for nearly a decade until the publication of SNMPv2. The saving grace for SNMPv1 was that most operators could hide their SNMPv1 implementations behind a firewall so that no outside entity could exploit the inherent weakness, however it still remained an issue as those connected to the internal trusted network could still easily discover the community strings.

SNMPv2 was built upon the framework of SNMPv1 by including additional management variables, additional security options, and several other improvements. However, the security implementation introduced in SNMPv2 turned out to be unwieldy and was ultimately rejected by the user community at large. Though SNMPv2 had new and improved management variables and new and improved ways to access those new variables, community strings still persisted. In fact the most common usage of SNMPv2 is SNMPv2c, the “c” connoting the use of community strings.

SNMPv3 is the long awaited answer to the problem of community strings. While SNMPv3 continues to build on the foundations laid by SNMPv1 and SNMPv2, it makes its significant contribution in the form of secure connections between manager and managed assets. For all practical purposes, security is the only enhancement issue SNMPv3 addresses; there are no other changes to the protocol [13]. That said, SNMPv3 also updates some of the terminology used when speaking about SNMP. While security was not a major concern for v1 and could not be agreed upon by the release of v2, SNMPv3’s single greatest contribution to the protocol was the implementation of a security scheme, which required proper authentication and authorization between NMS and Agent. Additionally SNMP3 no longer sends community names across the wire in the clear.

D. SNMP COMMUNICATIONS

At its base, SNMP is a means to transmit management messages over a network. Thus far, the discussion has only referred in a very generic manner either to the players in that exchange or to the actual management messages themselves. The two network entities involved in the exchange of messages are the SNMP Agent and the SNMP Network Management Station (NMS). The SNMP Agent is what actually sends and receives messages from the managed assets and the NMS is what the network administrator will interface with to manage said assets. The values that are exchanged and modified in SNMP are stored in the Management Information Base or MIB.

Since SNMP was designed with simplicity in mind the number of messages that can be exchanged between NMS and Agent is quite small. In SNMPv1 only five management messages were defined:

- Get – The Get message was sent from the NMS to the Agent to retrieve a specific value of a management variable monitored by the Agent.
- Set – The Set message was sent from the NMS to the Agent to modify a specific value of a management variable monitored by the Agent.
- Get-Next – The Get-Next command can be used to return a series of management variable values instead of just one.
- Get-Response – The Get-Response is the first of the Agent to NMS messages. This message is the response to either a Get or a Get-Next NMS request.
- Trap – Traps are automated messages sent from Agent to the NMS in response to some environment variable changing.

SNMPv2 introduced the following additional messages:

- GetBulk – GetBulk is a more efficient form of GetNext.
- Inform – The inform message was added to SNMPv2 to allow inter NMS communications. For the first time NMS could communicate with another NMS which provided the possibility of NMS hierarchies to manage large networks.

Though few in number, these messages are very powerful in that they: allow a network administrator to view almost any configuration or state variable on any number of devices, allow an administrator to remotely configure any number of devices, and provide automated alarming in the event of failures or the anticipation of failures. However, an administrator cannot be available to direct this process all the time. One of the greatest benefits of SNMP is that of automation. One of the greatest examples of this automation is trapping and polling.

Trapping and polling refer to the ways in which an NMS can keep an eye on the state of the network. Traps are sent when the value of a managed object goes outside a predefined threshold. It is important to note that the setting of thresholds is one of the most important aspects of managing a network with SNMP. If thresholds are set too high then faults could be regularly happening without any notification being sent to the NMS, rendering SNMP effectively useless. Set the thresholds too low and trap after trap after trap will be sent to the NMS, not only clogging up the network, but also creating a flood of information that then must be parsed to discover genuine problems. Network managers must take great care to set thresholds in that narrow window of generating not so few traps that faults are overlooked, and not so many traps that faults are lost in the deluge. In the polling approach, the NMS will regularly poll the Agent for relevant state information. This way, the NMS has a near real-time picture of the state of the whole network and the assets contained within it. Additionally, repeated polls over time can reveal patterns in the network that the NMS can then use to refine its polling process, allocating the largest resources to the most problematic areas of the network. The downside of polling comes in when an NMS becomes responsible for hundreds or thousands of assets. The constant polling messages sent out to keep up with the network can flood the network with traffic. This can not only overwhelm the NMS which constantly has to process requests and receipts of state information, but can exhaust network resources as well. A third option is Trap directed polling which uses the strengths of both Traps and Polling to offset their weaknesses. Each agent on the network has well-defined state variables that it is responsible for managing. Each of these variables has a corresponding threshold. In the event that a threshold is crossed, the Agent can send a trap message to the NMS to inform it that a significant event has occurred. This method has the advantage of not putting messaging traffic on the network unless there is an actual problem, and, unlike the polling method, the results are in real time, as the traps are sent off immediately after the detection of a problem. The disadvantage of this method is that it is costly to the agent, which often is a very small piece of code with limited resources. Also, if the thresholds are not tuned appropriately, the agent can flood the network with message after message. Another disadvantage to this method is that, should a networking device fail, the agent cannot then use the network to report to the NMS; in which case, the NMS is never notified of a potentially catastrophic failure on the network. In Trap directed polling, however, this is only half the story. To address the shortcomings of just trapping network events, Trap directed polling polls at a very low frequency to ensure that it can still communicate with every Agent and, as such, is not missing potential messages. This low frequency polling does not produce enough traffic to overwhelm the network or the NMS, and ensures that, if an Agent is unable to communicate a fault to the NMS, that the NMS becomes aware of the situation anyway. Additionally, in Trap directed polling, once a trap is received the NMS has the option to abandon the low frequency polling and start polling in a more probing fashion to those devices closest to the reported fault.

As mentioned previously, this all happens without the intervention of a network manager. Once the crucial step of setting up the Agent thresholds and the polling policy on the NMS has been completed a fair amount of automation is expected. Network administrators can even tie SNMP into SMS messages such that when faults or significant events happen on their networks, they can be notified via their cell phone or pager. Truly SNMP goes a long way toward supporting the goal of increased uptime in a network. Thus far, the discussion has been about how SNMP goes about exchanging messages. The following section will explore what the nature of those messages is as well as what information is passed in them.

In SNMPv1 the format for all of the messages was the same, except for the trap message. With the publication of SNMPv2, all SNMP messages were formatted to look exactly the same, only by reading the PDU type indicator, or command number, could an Agent or NMS distinguish the specific purpose of any particular SNMP message. As will be explained shortly the uniformity of SNMP turns out to be another one of its strengths. This uniformity is provided by the use of the Structure of Management Information (SMI).

While the SNMP protocol deals with the exchange of management messages, the messages must be exchanged by software entities. The SNMP Agent is the software entity on the managed asset and the Network Management Station (NMS) is the software entity installed on the centralized host a network administrator uses to manage the network. SNMP uses UDP an OSI Layer 4 transport protocol that is connectionless and relies on the application layer protocol to ensure delivery.

UDP is a connectionless, and therefore unreliable protocol; for example, it does not support acknowledgement. There is no built in capability for detecting and retransmitting lost packets; UDP datagrams are “fire and forget.” Using TCP, the connection-oriented and reliable Layer 4 transport protocol, might seem like a better idea at first glance however, there are some specific properties of UDP that make it a better fit. First of all, UDP is lightweight. Not only are the UDP datagrams smaller than TCP datagrams, but UDP datagrams will never require an acknowledgement or response. In a busy network where SNMP traffic is constantly going back and forth between Agents and the NMS, the small overhead of UDP makes it the better fit. If there was a network fault that severed communications, TCP would continually try to retransmit until it received an acknowledgement, possibly flooding the network. UDP, however, in the case of SNMP, would simply fire and forget the datagram, and leave it up to SNMP to decide how it wants to deal with a lack of response (which would be to wait for a specific amount of time and then transmit another UDP datagram). Both the Agent and the NMS of SNMP live at the top, or Application layer of the OSI model. SNMP uses UDP port 161 for polling and port 162 for traps.

1. MIB

MIBs are specifications containing definitions of management information so that networked systems can be remotely monitored, configured, and controlled [14]. They can be thought of as the database of system variables that can be monitored or configured by SNMP. The framework of that database is known as the Structure of Management Information (SMI), which uses the Abstract Syntax Notation One (ASN.1) to describe and define what variables can be monitored, how those variables are stored, and the syntax for messages communicating the status or seeking to modify those variables. If

ASN.1 were a language (which to an extent it is) SMI would be the grammar rules used to create MIBs. SMI, using a subset of ASN.1, defines the allowed data types in a MIB, the rules for specifying object types, and the rules for defining the events that trigger an SNMP response. This common language and common grammar for the creation of MIBs and MIB variables is extremely powerful in that it allows anyone to create a MIB that is then interoperable with any SNMP implementation. From this framework, any number of variables can be defined for any number of different assets be they hardware or software.

All of the various types of management information that can be stored in a MIB are called objects. Any specific instance of an object is called a variable, and it is the values contained in the variables that SNMP monitors and modifies. Objects are defined by their data types, access type, identity, behaviors and how individual instances of the object are identified. There are two types of objects in the SNMP MIB: the scalar object and the columnar object. Columnar objects can have zero or more variables stored in them at any one time and are stored in a logical data table as defined by SMI. Individual instances are identified by an indexing scheme defined for that object type. An example of a columnar object would be the instances of the physical ports on a networking device, as there may be between two and several dozen. Scalar objects, on the other hand, are objects of which there can be one and only one possible instance. For example, “system uptime since last reboot” refers to a constant count of seconds elapsed since the device was last rebooted, for which there can be only one value. There is no index required for these instances, as they are unique. Any and all SNMP objects are divided up between these two types of object; however, to identify any particular MIB variable one must look to the Object Identifier (OID).

In order for the panoply of existing MIB objects to be quickly referenced, a hierarchical numbering and naming scheme was devised to track them. The naming scheme is simply an extension of the numbering scheme to make the long chains of numbers that make a particular OID human readable. As shown in the diagram below, the location of any particular MIB object can be located by using the numbers in its tree; this value is also known as “the identity of the object.” Only the objects at the bottom of the tree (also known as a leaf) can contain values. So the OID 1.3.6.1.2.1 or iso.org.dod.internet.mgmt.mib-2 contains no values to query or configure, but the OID 1.3.6.1.2.1.2.2 or iso.org.dod.internet.mgmt.mib-2.interfaces.2 would. It is also worth noting that for the last example the final number connotes the second instance of interface, the next entry would be 1.3.6.1.2.1.2.3 and so on. For scalar objects, the final numeral value is always 0 as there are not multiple instances to enumerate. For example, the OID for system uptime is 1.3.6.1.2.1.1.3.0 or iso.org.dod.internet.mgmt.mib-2.system.sysUptime.0. Thus far the examples used have all come from 1(iso).3(org).6(dod).1(internet).2(mgmt).1(mib-2); the reason is that the mib-2 subtree has special significance in SNMP. MIB-II, as it is formally known, is currently the minimum standard of variables any system using SNMP must use. Every device using SNMP must use MIB-II. MIB-II contains the following subtrees:

```
-- groups in MIB-II

          system       OBJECT IDENTIFIER ::= { mib-2 1 }

          interfaces   OBJECT IDENTIFIER ::= { mib-2 2 }

          at           OBJECT IDENTIFIER ::= { mib-2 3 }

          ip           OBJECT IDENTIFIER ::= { mib-2 4 }

          icmp         OBJECT IDENTIFIER ::= { mib-2 5 }

          tcp          OBJECT IDENTIFIER ::= { mib-2 6 }

          udp          OBJECT IDENTIFIER ::= { mib-2 7 }

          egp          OBJECT IDENTIFIER ::= { mib-2 8 }

          -- historical (some say hysterical)
          -- cmot      OBJECT IDENTIFIER ::= { mib-2 9 }

          transmission OBJECT IDENTIFIER ::= { mib-2 10 }

          snmp         OBJECT IDENTIFIER ::= { mib-2 11 }[15]
```

Other than MIB-II, any particular implementation of SNMP is free to use whatever other MIBs are appropriate. As mentioned earlier, MIBs can be defined by anyone making them indefinitely extensible. Another important subtree to know is the private subtree located at 1.3.6.1.4 or iso.org.dod.internet.private. This subtree contains all of the privately developed MIBs created by individuals, institutions, organizations, companies, etc. to manage a particular asset. Currently the list of contributions to the private tree is registered by the Internet Assigned Numbers Authority (IANA). For example Cisco product MIBs are all located under 1.3.6.1.4.1.9 or iso.org.dod.internet.private.enterprises.cisco subtree. This nature of MIBs to be created in a standard format and created by virtually anyone is one of SNMPs greatest strengths. Upon purchase of a new Cisco device, one can download the relevant MIBs created by Cisco and simply incorporate them into the existing management framework. At which time a network administrator can enjoy the numerous gains offered by the device-specific MIB. It's important to note that both the management station and the Agent residing on a managed asset have to have the same MIBs to work from. A management station must load the same MIBs that the Agent is using in order to view or configure the variables in that MIB.

In addition to OID, and whether or not an object is scalar or columnar, SMI also defines, for each object type, the various types of data that the object can hold. Like SNMP itself, SMI has gone through revisions. Accordingly, SMI for SNMPv1 was called "SMIv1," and had only ten data types available to it. The second version of SMI, SMIv2, was published in response to the shortcomings identified by the user community who had worked with SNMPv1 as defined by SMIv1. SMIv2 added new data types, and, as it is the current standard, the data type discussion will be limited to the following:

- Integer32 – Any positive or negative number. The maximum value of Integer32 is +/- 2^{31}-1 or +/- 2,147,483,647.
- INTEGER (enumerated) – In this instance of INTEGER the values associated with either positive or negative numbers are assigned a label. For example, the status of a port on a network device could be 1(up), 2(down), 3(unknown).
- Unsigned32 – This data type contains non-negative numbers from 0 to 2^{32}-1.
- Gauge32 – This data type is synonymous with Unsigned32 as it also contains non-negative numbers from 0 to 2^{32}-1. However, Gauge32 does have a specific behavior defined that states that Gauge32 will only return values within its range regardless of the actual value. This is analogous to a thermometer with a range of 0^o to 100^o showing 100^o even if the temperature is 112^o.

- Counter32 – The Counter32 data type is a non-negative integer from 0 to 2^{32}. This data type differs from Gauge32 in that Gauge32 is meant to fluctuate up and down and Counter32 is not. Counter32 increments upwards until it reaches the upper limit of 2^{32} and then wraps back to the original starting number (which can be defined as non-zero).
- Counter64 – Counter64 is the same as Counter32 except that its maximum possible value is 2^{64}.
- TimeTicks – This data type is used to count the number of hundredths of a second in between two events. The maximum value of TimeTicks is 2^{32}, which is roughly equivalent to 497 days at which time the TimeTicks counter will wrap back to a starting value of 0.
- OCTET STRING – This data type specifies octets of binary or textual information. Using ASCII which represents eight binary bits (an octet) as an alpha character (A,B,C,etc.) this fields can store text. Additionally, OCTET STRING can store IP or MAC addresses as both can be represented in octet strings.
- OBJECT IDENTIFIER – This data type is synonymous with the numerical OID number that references an object in the hierarchical OID tree.
- Ip Address – This data type of four octets was designed to store IPv4 values.
- Opaque – This data type specifies octets of binary information. While there was no limit to this field in SMIv1, SMIv2 specifies an upper limit of 2^{16}-1 octets.
- BITS – The BITS data type, like INTEGER specifies an enumeration of labeled bits.

This concludes the list of definition possibilities for the creation of object types in SMIv2 for SNMP. All modern MIBs are a collection of objects defined by the above rules. Now that the MIBs have been explained, the next part of the puzzle is to look at how the Agent and the NMS work together to manage the collection of objects.

E. HOW SNMP WORKS

Now that all of the pieces have been identified, here is an example of the process of SNMP in use. Firstly, the end user of an NMS enters a command to retrieve some value from a managed asset (although there certainly does not need to be an end user involved, as many NMS functions are automated). The NMS then creates the SNMP message with a Request ID, command number distinguishing the various request types, community name, and SNMP PDU. SNMP, being a layer 7 protocol, then hands off the message to the other layers of the OSI stack for processing. The message is ultimately sent over UDP port 161 to the receiving host.

When an Agent is enabled on a remote host device, in addition to managing the various local MIB variables, a primary function of the agent is to listen for an incoming request from an NMS. Upon receipt of such a message the first thing that the agent does is to check for proper ASN.1 formatting. If this check fails, the message is discarded with no notification. Should it not fail, the next step is to check the SNMP version number. If the version number is incorrect, the message is discarded without notification. If the SNMP message passes the first two checks, the next information to be checked is the community string. This is the first time whereby, if a failure occurs, a message is sent back to the originating NMS. At this point, any further failures of the message will generate notification back to the NMS about the reason for the failure. If, however, the community string name matches, then the PDU is decoded and processed. It is important to note that if the processing of the message fails due to lack of requested variables, etc., a message is sent back to the NMS giving a generic reason for the failure. When the MIB information requested is finished being collected, the Agent then creates a response message. The message contains command number 2 for Get-Response and the same request ID as the NMS sent; this allows the NMS to track all incoming Get-Response messages. After the message is created, it is put back on the wire and sent back to the NMS for processing, at which point the Agent returns to its default listening state.

F. FCAPS REVISITED

The original conversation about FCAPS was in terms of what FCAPS was and the responsibilities of each of the sub divisions. The following section will describe SNMP’s contributions to each of the components of the FCAPS model.

- Fault Management – through the methods of polling and trapping, SNMP can provide a near real-time picture of what devices are, and, more importantly, what devices are not operating properly on the network. In the event of a fault, traps can be used to immediately notify a network manager via a console or even a cell phone. In the event that the Trap fails, the NMS can use a backup low frequency polling method to detect faults where no traps were received. SNMP can also be used to help isolate faults from affecting the rest of the network in addition to providing alarm correlation capabilities. The concept of high availability is about ensuring that services provided by IT are available whenever an end user needs them. This is obviously important in the case of a soldier relying on information from an IEEE Std. 802.16 BS. Along with reducing single points of failure, one of the key aspects of high availability is fault detection. After all, the sooner a fault is detected the sooner it can be resolved. Though in the early days of SNMP this Fault Management capability was limited to routers and switches, now the capability has expanded to include UPS, printers, workstations, and software. With this increased functionality, SNMP is no longer limited to monitoring network-support devices only (as the original protocol was designed to do), but is now capable of monitoring every device connected to the network as well.
- Configuration Management – SNMP provides configuration management on several levels. First off, by using ICMP messages (more commonly known as “ping” messages), many current NMS software suites can perform a discovery of the network creating a logical topology of the network. This initial mapping in combination with trapping and polling can be used as a baseline network level configuration in which the NMS can detect and record changes. SNMP can also detect and record changes on the device level as well. As SNMP is built into the initial distributions of more and more hardware and software assets, increasingly

there are fewer and fewer either configuration or state variables an asset can have that can’t be managed by SNMP. As these variables change, traps can be sent to the NMS such that any change either on the network level or the device level can be logged.

- Accounting Management – with traps designed to fire at user logon, SNMP can track total times a user is logged on, and total network resources consumed by any particular user. Current market products such as Cisco’s NAC (Network Access Control) product line rely on SNMP to either permit or deny an end user access to the network.
- Performance Management – SNMP, using its combination of polling and trapping, can monitor for bottlenecks and track usage trends for capacity planning. This capacity planning in turn allows for intelligent preventative maintenance. In addition to merely tracking traffic patterns, SNMP can track operating temperatures and CPU usage to not only give a picture of how any given device is performing, but the network as a whole as well.
- Security Management – as mentioned, Cisco’s NAC product line, after receiving authentication information from a user, uses SNMP to enable the physical port the user is connected to, or to disable it, thereby preventing access to the network. SNMP can increase the security posture of any network by tracking user statistics, and providing a wealth of information designed to keep the network operating at peak efficiency and reducing costly down times.

As should be obvious by now, the benefits of SNMP are numerous to say the least. It is lightweight, ubiquitous in the networking field, and anyone can contribute to the wealth of functionality already attributed to it. These benefits have not gone unnoticed by the IEEE 802.16 working group, as they have included a MIB as an amendment to the standard.

IV. 802.16F

In 2005, the 802.16 working group published IEEE Std. 802.16f, which is an amendment to the IEEE 802.16 standard and which introduces a default MIB for use in IEEE Std. 802.16 implementations. Up until the release of 802.16f, different vendors were free to use or create whatever MIB they felt appropriate for their individual implementations. This type of vendor specific usage of MIBs can lead to interoperability issues as well as consumer headaches when choosing a specific vendor's implementation.

The IEEE Std. 802.16 MIB contains two MIB modules: wmanIfMib and wmanDevMib. The wmanIfMib module defines the managed objects germane to the IEEE Std. 802.16 interface. The wmanDevMib module, on the other hand, deals with the managed objects germane to devices that implement the IEEE Std. 802.16 interface. Each of these two MIB modules is further broken down into three areas, each dealing with the managed objects contained in the SS and BS, respectively, as well as one area for common objects.

wmanIfMib

- wmanIfBsObjects
- wmanIfSsObjects
- wmanIfCommonObjects

wmanDevMib

- wmanDevBsObjects
- wmanDevSsObjects
- wmanDevCommonObjects

The remainder of this chapter will explore in detail the hierarchy of managed objects in these two MIB modules. However, before the conversation about specific MIB values begins, it is important to understand how a data abstraction known as a Textual Convention can be used to increase the number of possible data types available to objects.

Textual Conventions are definitions that can be included into a MIB to make programming the MIB significantly easier. As mentioned previously, object types can only be configured to hold one of several data types. (INTEGER, Counter32, Gauge32, OCTET STRING, etc.) Textual Conventions are a way to manipulate these “primitive” data types into more useful abstractions. For example, a very common use of a Textual Convention is that of Display String as defined in MIB-II:

```
DisplayString ::=
              OCTET STRING
          -- This data type is used to model textual information taken
          -- from the NVT ASCII character set.  By convention, objects
          -- with this syntax are declared as having

          --
     --    SIZE (0.255) [15]
```

Note that none of the primitive data types can hold alpha characters, that is, human readable letters of the alphabet. To deal with this shortcoming, the Display String Textual Convention uses the ASCII character set (a character set that uses 8 bits or 1 byte of data to represent an alpha character) in combination with the OCTET STRING data type to define a way to insert human readable alpha characters into a MIB. Textual Conventions are extremely useful in that once defined, they allow a programmer to create just about any data type imaginable, so long as it can be derived from the 12 original data types. Below is an example of a Textual Convention specific to 802.16f:

```
WmanIfIpVersionType ::= TEXTUAL-CONVENTION
        STATUS        current
        DESCRIPTION
                "The object of this type indicates the version of IP used
                on the Secondary Management Connection. The values should
                be undefined if the 2nd management CID doesn't exist."
        REFERENCE
                "Subclause 11.7.4 in IEEE Std 802.16-2004"
        SYNTAX        INTEGER        {undefined(0),
                                     ipv4(1),
                                     ipv6(2)}
```

This example illustrates how the WmanIfIpVersionType Textual Convention defines how the values of INTEGER can be used to convey the three possible version states of an IP connection. It is important to note that all SNMP communications work in this fashion.

Should an administrator decide that they want to shut down an interface on a networking device, the control message (a SET message) that goes from the NMS to the Agent is not “shut down port x.” In the case of a Set command, the control message passes a numerical value which, through the use of the SMI definitions and the Textual Convention definitions, can be construed by the Agent as a prompt for a specific action. In the case of a Get command (for the version of IP being used in this case), the NMS would query the object that uses WmanIfIpVersionType as its SYNTAX type. Were the IP connection IPv4, the Agent would return the value “1,” which would then be interpreted by the corresponding MIB on the NMS to a value of IPv4. This is just another way in which SNMP helps to keep the traffic being generated lightweight; instead of having to pass human readable values, it can pass simple numerical values which are then interpreted using SMI and implementation specific Textual Conventions.

A. WMANIFMIB

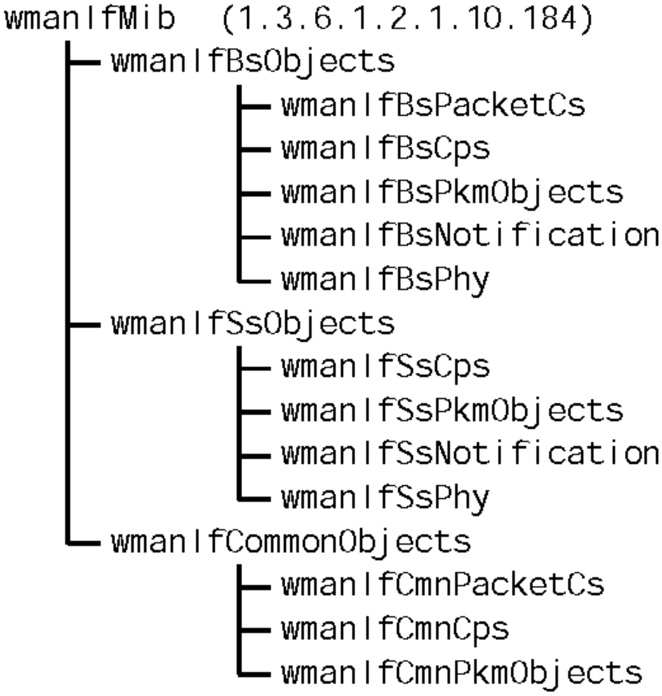

Figure 9. wmanIfMib Structure

The first of the MIB modules defined in 802.16f is wmanIfMib, which defines management objects relevant to the IEEE Std. 802.16 broadband wireless interface. The figure above illustrates a high level view of the MIB. The wmanIfMib, like the wmanDevMib module, is further broken down into three subtrees, the first of these being wmanIfBsObjects.

1. wmanIfBsObjects

wmanIfBsObjects, though under the wmanIfMib, is still a subtree and not a leaf. As such, it holds no object data of its own. Instead, it holds the next subtree, wmanIfBsPacketCs.

The following object groups are defined for use in all BS:

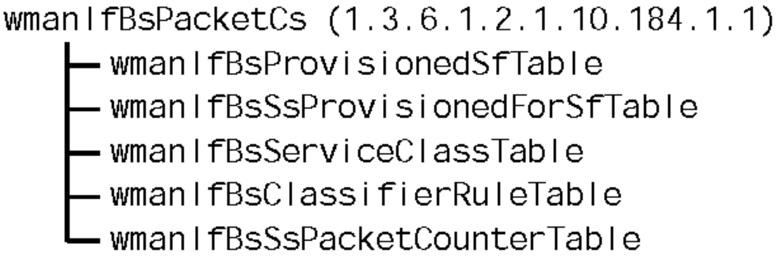
wmanIfBsPacketCs (1.3.6.1.2.1.10.184.1.1)
wmanIfBsProvisionedSfTable
wmanIfBsSsProvisionedForSfTable
wmanIfBsServiceClassTable
wmanIfBsClassifierRuleTable
wmanIfBsSsPacketCounterTable

Figure 10. wmanIfBsPacketCs Structure

wmanIfBsPacketCs, as a subtree itself, holds the leaf values as shown in Figure 11. This subgroup of leaves contains values germane to the Packet CS Management entity layer, which deals with the Convergence Sublayer of IEEE Std. 802.16 which prepares layer three protocol traffic for the IEEE Std. 802.16 MAC layer.

<u>wmanIfBsProvisionedSfTable</u>

This columnar object contains provisioned service flow profiles for SS. Its SYNTAX (data type) is a sequence (or rows) of objects wmanIfBsProvisionedEntry, which is itself a listing of object variables pertinent to the state of a service flow of a SS.

<u>wmanIfBsProvisionedForSfTable</u>

This object maps the MAC address of an SS to a service flow. This can be used to enable multicast where multiple MAC addresses are mapped to the same service flow.

<u>wmanIfBsServiceClassTable</u>

This object contains QoS relevant parameters by using a sequence of objects, wmanIfBsServiceClassEntry, each of which itself lists specific QoS values, such as allowed jitter or latency.

<u>wmanIfBsClassifierRuleTable</u>

This object uses a sequence of wmainIfBsClassfierRuleEntry to index classification rules for service flows. This table, along with the Service Class table, are both referenced by the Provisioned Service Flow table above to create a complete picture of packet QoS, and service flow association from CS to the CPS.

<u>wmanIfBsSsPacketCounterTable</u>

This object tracks the number of packets in and out of any particular service flow. It uses a sequence of the object, wmanIfBsSsPacketCounterEntry, which itself is a list of several counters which record the number of octets sent or received by the BS per service flow

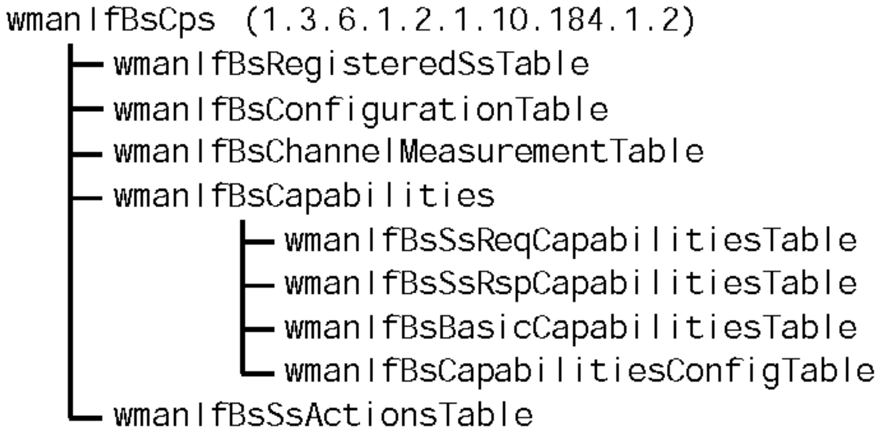
wmanIfBsCps (1.3.6.1.2.1.10.184.1.2)
wmanIfBsRegisteredSsTable
wmanIfBsConfigurationTable
wmanIfBsChannelMeasurementTable
wmanIfBsCapabilities
wmanIfBsSsReqCapabilitiesTable
wmanIfBsSsRspCapabilitiesTable
wmanIfBsBasicCapabilitiesTable
wmanIfBsCapabilitiesConfigTable
wmanIfBsSsActionsTable

Figure 11. wmanIfBsCps Structure

The wmanIfBsCps MIB subtree deals with objects related to the CPS layer of the BSMAC implementation.

wmanIfBsRegisteredSsTable

As the name implies, this object tracks SS which have registered with the BS through the RNG-REQ and RNG-RSP process. It uses a sequence of WmanIfBsRegisteredSsEntry to list such connection variables as MAC address and Primary and Secondary CID values.

wmanIfBsConfigurationTable

This table holds configuration information about the BS default behavior for scheduling a second management channel as well as for the CPS scheduler for Adaptive Antenna Systems(AAS). This object uses a sequence of the wmanIfBsConfigurationEntry object, which itself is a list of objects that define such parameters as UCD and DCD spacing as well as DL/UL-MAP spacing

wmanIfBsChannelMeasurementTable

This table keeps a record of SS to BS signal strength variables recorded during the RNGREQ and RNG-RSP process. This object uses a sequence of WmanIfBsChannelMeasurementEntry, which is itself a list of signal quality measurements.

wmanIfBsCapabilities

This subtree contains objects which record the SS and BS capabilities as discovered from the RNG-REQ and RNG-RSP process.

wmanIfBsSsReqCapabilitiesTable

This table contains a sequence of WmanIfBsSsReqCapabilitiesEntry which, for each entry, defines the SSs capabilities as per the RNG-REQ and RNG-RSP process.

wmanIfBsSsRspCapabilitiesTable

This table contains a record of the capabilities used by BS and SS as negotiated by the REQ and RNG-RSP process.

wmanIfBsBasicCapabilitiesTable

This table, like the wmanIfBsSsReqCapabilitiesTable, defines the capabilities available to the BS as defined by BS hardware and software. It is the information combined in these two tables that makes up the capability negotiation framework.

wmanIfBsCapabilitiesConfigTable

A BS has a certain number of capabilities out of the box. This object is used to restrict the number of raw capabilities that the BS is capable of for compliance with certain regulatory requirements where applicable.

wmanIfBsSsActionsTable

This table also records values relating to regulatory restrictions; however it refers to the SS instead of the BS. This table contains values regarding what and how the SS is allowed to transmit.

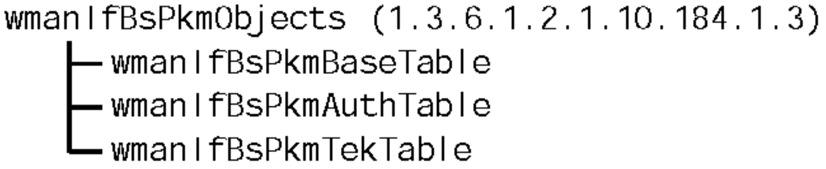

Figure 12. wmanIfBsPdmObjects Structure

wmanIfBsPkmObjects

The wmanIfBsPkmObjects subtree contains objects germane to the Security Sublayer in IEEE Std. 802.16 and managed by the BS.

wmanIfBsPkmBaseTable

This table contains a sequence of occurrences of object, wmanIfBsPkmBaseEntry, which, per SS, lists PKM variables such as AK and TEK lifetimes, as well as counts of authorization/ authentication requests and replies.

wmanIfBsSsPkmAuthTable

This table tracks a more robust set of per SS authentication and authorization variables. This object tracks the various SAs between BS and SS.

wmanIfBsPkmTekTable

This table tracks the array of TEKs distributed to the various SS. Example variables are Security Association ID (SAID) and the data encryption algorithm used per TEK.

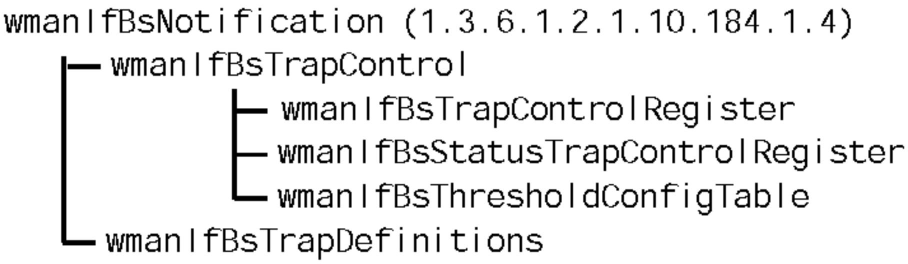
wmanIfBsNotification (1.3.6.1.2.1.10.184.1.4)
wmanIfBsTrapControl
wmanIfBsTrapControlRegister
wmanIfBsStatusTrapControlRegister
wmanIfBsThresholdConfigTable
wmanIfBsTrapDefinitions

Figure 13. wmanIfBsNotification Structure wmanIfBsNotification

This subtree breaks from the pattern thus far, and does not contain any objects referencing the nature of any specific connections between BS and SS, but instead defines the types of notification that can be used in IEEE Std. 802.16.

wmanIfBsTrapControl

This subtree contains objects defining what traps can be sent back to an NMS for error reporting.

wmanIfBsTrapControlRegister

This first leaf object under wmanIfBsNotification defines what BS traps are enabled for reporting to the NMS.

wmanIfBsStatusTrapControlRegister

This object defines status notifications from the BS to the NMS. It differs from the above trap notifications in that these messages are not about what traps are allowed, but instead, these traps communicate information about connection success or failure events.

wmanIfBsThresholdConfigTable

This table allows a network administrator to set integer thresholds in units of dBm (decibels per milliwatt) of received signal strength. This Received Signal Strength Indication (RSSI) threshold, once crossed, then generates a traps back to the NMS.

wmanIfBsTrapDefinitions

This object contains all possible notification and trap types that a BS can receive from an SS.

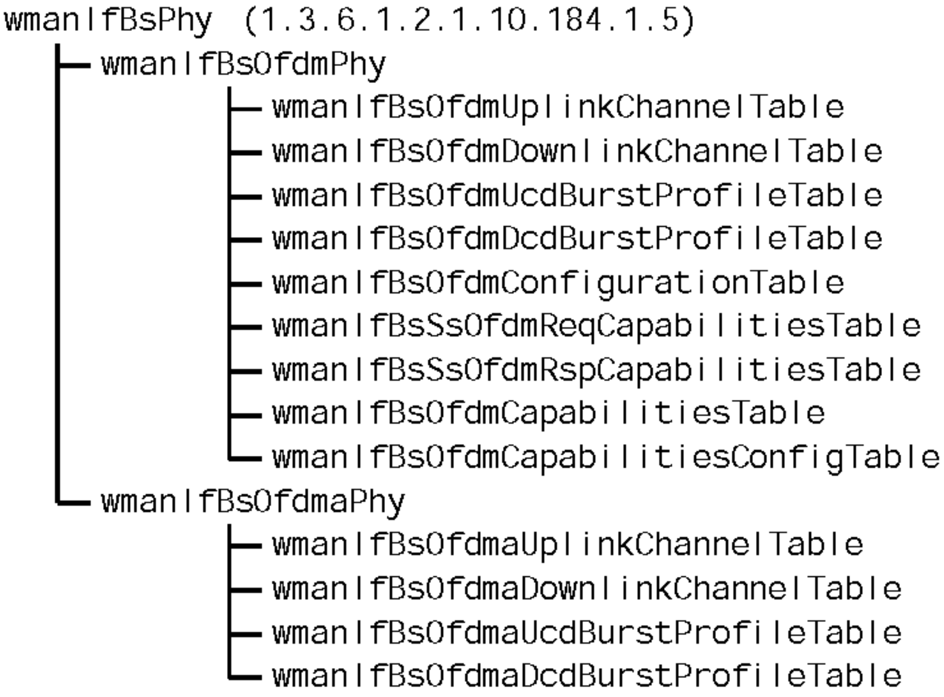
wmanIfBsPhy (1.3.6.1.2.1.10.184.1.5)
wmanIfBsOfdmPhy
wmanIfBsOfdmUplinkChannelTable
wmanIfBsOfdmDownlinkChannelTable
wmanIfBsOfdmUcdBurstProfileTable
wmanIfBsOfdmDcdBurstProfileTable
wmanIfBsOfdmConfigurationTable
wmanIfBsSsOfdmReqCapabilitiesTable
wmanIfBsSsOfdmRspCapabilitiesTable
wmanIfBsOfdmCapabilitiesTable
wmanIfBsOfdmCapabilitiesConfigTable
wmanIfBsOfdmaPhy
wmanIfBsOfdmaUplinkChannelTable
wmanIfBsOfdmaDownlinkChannelTable
wmanIfBsOfdmaUcdBurstProfileTable
wmanIfBsOfdmaDcdBurstProfileTable

Figure 14. wmanIfBsPhy Structure

wmanIfBsPhy

The wmanIfBsPHy subtree contains the last of the BS specific interface subtrees, one for OFDM and one for OFDMA. Without belaboring the point, every leaf under this subtree defines a physical characteristic of the channels to and from BS and SS. The variables make specific reference to DL/UL frequency values, DCD/UCD values, frame durations, burst profiles, modulations etc. Whereas the previous objects varied in the nature of what they defined, the objects in this subtree all define the physical connection characteristics between BS and SS.

2. wmanIfSsObjects

The second of the subtrees under the wmanIfMib module is the wmanIfSsObjects subtree. This subtree refers specifically to managed objects residing in the SS. Whereas every BS must implement the 802.16f MIB as per the standard, only a managed SS is required to enable the MIB. If, however, the SS is a managed device and the MIB is implemented in full, the high-level view of the SS Interface Objects MIB looks identical to the BS Interface Objects MIB, with the notable exception of a Packet CS object. As all service flows are defined at, and managed by, the BS, the subtrees and objects associated with the creation and management of these variables are not needed in the SS's implementation of its MIB. Another difference between the BS and SS subtrees is that the BS objects need to at any time be able to reference the sum of all SS that are connected while the SS need only define its unique connection with one BS. Below is an example of how the same high-level object can look different upon close inspection:

```
wmanIfBsConfigurationEntry OBJECT-TYPE
        SYNTAX WmanIfBsConfigurationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
                "This table is indexed by ifIndex with
                an ifType of propBWAp2Mp."
        INDEX { ifIndex }
        ::= { wmanIfBsConfigurationTable 1 }

WmanIfBsConfigurationEntry ::= SEQUENCE {
        wmanIfBsDcdInterval                  INTEGER,
        wmanIfBsUcdInterval                  INTEGER,
        wmanIfBsUcdTransition                INTEGER,
        wmanIfBsDcdTransition                INTEGER,
        wmanIfBsInitialRangingInterval       INTEGER,
        wmanIfBsSsULMapProcTime              Unsigned32,
        wmanIfBsSsRangRespProcTime           Unsigned32,
        wmanIfBsT5Timeout                    INTEGER,
        wmanIfBsT9Timeout                    INTEGER,
        wmanIfBsT13Timeout                   INTEGER,
        wmanIfBsT15Timeout                   INTEGER,
        wmanIfBsT17Timeout                   INTEGER,
        wmanIfBsT27IdleTimer                 Unsigned32,
        wmanIfBsT27ActiveTimer               Unsigned32,
        wmanIfBs2ndMgmtDlQoSProfileIndex     INTEGER,
        wmanIfBs2ndMgmtUlQoSProfileIndex     INTEGER,
        wmanIfBsAutoSfidEnabled              INTEGER,
        wmanIfBsAutoSfidRangeMin             Unsigned32,
        wmanIfBsAutoSfidRangeMax             Unsigned32,
        wmanIfBsAasChanFbckReqFreq           INTEGER,
        wmanIfBsAasBeamSelectFreq            INTEGER,
        wmanIfBsAasChanFbckReqResolution     INTEGER,
        wmanIfBsAasBeamReqResolution         INTEGER,
        wmanIfBsAasNumOptDiversityZones      INTEGER,
        wmanIfBsResetSector                  INTEGER}
```

```
wmanIfSsConfigurationEntry OBJECT-TYPE
        SYNTAX WmanIfSsConfigurationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
                "This table is indexed by ifIndex."
        INDEX { ifIndex }
        ::= { wmanIfSsConfigurationTable 1 }

WmanIfSsConfigurationEntry ::= SEQUENCE {
        wmanIfSsLostDLMapInterval            INTEGER,
        wmanIfSsLostULMapInterval            INTEGER,
        wmanIfSsContentionRangRetries        INTEGER,
        wmanIfSsRequestRetries               INTEGER,
        wmanIfSsRegRequestRetries            INTEGER,
        wmanIfSsTftpBackoffStart             INTEGER,
        wmanIfSsTftpBackoffEnd               INTEGER,
        wmanIfSsTftpRequestRetries           INTEGER,
        wmanIfSsTftpDownloadRetries          INTEGER,
        wmanIfSsTftpWait                     INTEGER,
        wmanIfSsToDRetries                   INTEGER,
        wmanIfSsToDRetryPeriod               INTEGER,
        wmanIfSsT1Timeout                    INTEGER,
        wmanIfSsT2Timeout                    INTEGER,
        wmanIfSsT3Timeout                    INTEGER,
        wmanIfSsT4Timeout                    INTEGER,
        wmanIfSsT6Timeout                    INTEGER,
        wmanIfSsT12Timeout                   INTEGER,
        wmanIfSsT14Timeout                   INTEGER,
        wmanIfSsT16Timeout                   INTEGER,
        wmanIfSsT18Timeout                   INTEGER,
        wmanIfSsT19Timeout                   INTEGER,
        wmanIfSsT20Timeout                   INTEGER,
        wmanIfSsT21Timeout                   INTEGER,
        wmanIfSsSBCRequestRetries            INTEGER,
        wmanIfSsTftpCpltRetries              INTEGER,
        wmanIfSsT26Timeout                   INTEGER,
        wmanIfSsDLManagProcTime              INTEGER}
```

Figure 15. Comparison of BS and SS Object Type Configuration Entry

Even a cursory glance reveals the significant difference in data types and number of objects referenced in the BS and SS Configuration Entry object. It is important to note these differences and it is why it cannot simply be assumed that because two subtrees look identical that they in fact are identical.

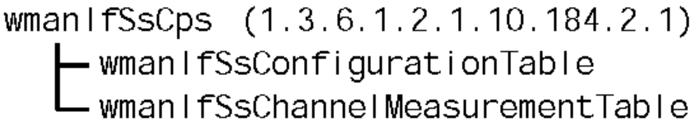
wmanIfSsCps (1.3.6.1.2.1.10.184.2.1)
wmanIfSsConfigurationTable
wmanIfSsChannelMeasurementTable

Figure 16. wimanIfSsCps structure

wmanIfSsCps

wmanIfCps is the first subtree under wmanIfSsObjects and, like its BS, has no object variables. This subtree contains the two leaves that define the CPS layer configuration information for the various connections between BS and SS.

wmanIfSsConfigurationTable

This table, unlike its BS counterpart, only contains one row. The object variables are stored in a sequence of wmanIfSsConfigurationEntry (as illustrated on the previous page), which itself is a listing of pertinent connection variables for the SS.

wmanIfSsChannelMeasurementTable

This table tracks through the use of a histogram reflecting RSSI and Carrier to Interference-plus-Noise-Ratio (CINR) signal quality measurements from BS to SS.

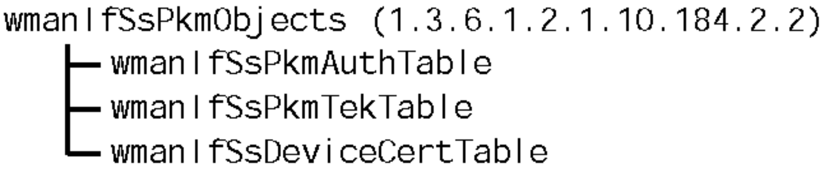

Figure 17. wmanIfSsPkmObjects Structure

wmanIfSsPkmObjects

This subtree, as does its BS counterpart, defines connection objects germane to the Security Sublayer for the IEEE Std. 802.16 stack.

wmanIfSsPkmAuthTable

As mentioned previously, it is the job of the SS to authenticate itself to the BS, and the BS does not need to authenticate itself to any SS. Thus, this table contains information regarding the authentication process from SS to BS. Information such as key lifetimes and authentication error codes are stored here.

wmanIfSsPkmTekTable

This table contains one row per TEK. Each SAID spawns uniquely one TEK, and everything from the encryption method used in the TEK to the key lifetime of the TEK are stored here.

wmanIfSsDeviceCertTable

This table contains the values of the X.509 SS certificate and manufacturers certificate per SS interface.

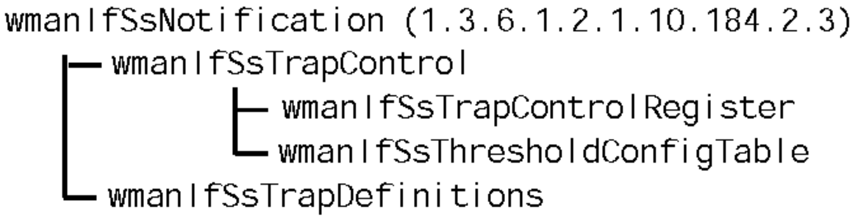
wmanIfSsNotification (1.3.6.1.2.1.10.184.2.3)
wmanIfSsTrapControl
wmanIfSsTrapControlRegister
wmanIfSsThresholdConfigTable
wmanIfSsTrapDefinitions

Figure 18. wmanIfSsNotification Structure

wmanIfSsNotification

The wmanIfSsNotification subtree mirrors the wmanIfBsNotification subtree except that, instead of the variables and values residing on the BS, they reside on the SS.

wmanIfSsTrapControl

This subtree groups all of the notifications possible for any SS.

wmanifSlTrapControlRegister

This object is used to activate or deactivate various levels of notifications (traps) for the SS. While similar to the BS version, the SS version of this object contains no references to PKM or Registration events wmanIfSsThresholdConfigTable

While it is more common for the BS and SS objects to have different objects and data types defined for them, this object is an exception in that the Threshold Config table measured the same RSSI values and even uses the same data types to record them.

wmanIfSsTrapDefinitions

As its name implies, this object contains a list of all of the objects available to the Control Register table for notification.

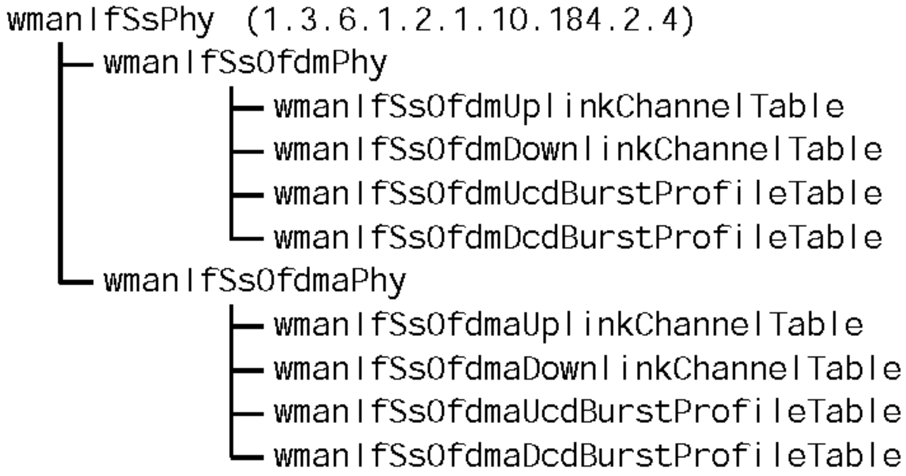
wmanIfSsPhy (1.3.6.1.2.1.10.184.2.4)
wmanIfSsOfdmPhy
wmanIfSsOfdmUplinkChannelTable
wmanIfSsOfdmDownlinkChannelTable
wmanIfSsOfdmUcdBurstProfileTable
wmanIfSsOfdmDcdBurstProfileTable
wmanIfSsOfdmaPhy
wmanIfSsOfdmaUplinkChannelTable
wmanIfSsOfdmaDownlinkChannelTable
wmanIfSsOfdmaUcdBurstProfileTable
wmanIfSsOfdmaDcdBurstProfileTable

Figure 19. wmanIfSsPhy Structure

wmanIfSsPhy

The wmanIfSsPhy subtree, like the PHY subtree for the BS, defines the characteristics of the physical connection between SS and BS. Again, objects here define such characteristics as UCD and DCD as well as frequency ranges and frame lengths. This subtree is also one of the few subtrees that resemble the BS subtree upon closer inspection as it describes the same global objects, just from different ends.

3. wmanIfCommonObjects

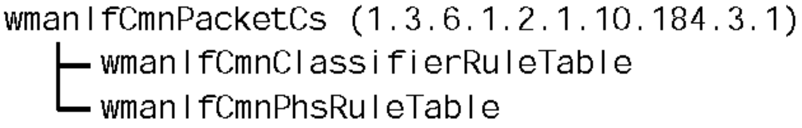
wmanIfCmnPacketCs (1.3.6.1.2.1.10.184.3.1)
wmanIfCmnClassifierRuleTable
wmanIfCmnPhsRuleTable

Figure 20. wmanIfCmnPacketCs Structure

The wmanIfCommonObjects subtree contains objects that are germane to session properties without any bias to whether they are on a BS or SS. These objects describe global conditions of the connection which both SS and BS must maintain.

wmanIfCmnPacketCs

This subtree defines CS objects common to both the SS and BS.

wmanIfCmnClassifierRuleTable

For each Service Flow that is opened from the BS to SS, there are a number of relevant parameters for that Service Flow. The rows in the Classifier Rule Table define each of the service flows in terms of IP addresses, priority, MAC addresses, source/destination ports, etc.

wmanIfCmnPhsRuleTable

In some applications, Voice over IP (VoIP) for instance, a number of VoIP PDUs may be put into one Layer 2 frame. If this is the case, then all of the header information for those packets (Source Address, Destination Address, QoS parameters) will be redundant. In order to lessen overhead associated with such communications, Payload Header Suppression groups a number of packets under one header. The PHS Rule Table keeps track of the data streams using PHS and what service flows are associated with what PHS rules on both ends of the connection to ensure that PHS packets arrive when and where they need to.

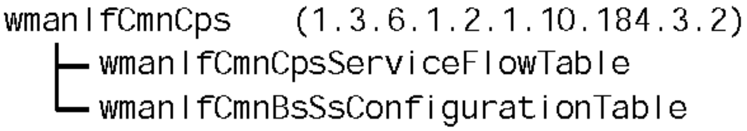
wmanIfCmnCps (1.3.6.1.2.1.10.184.3.2)
wmanIfCmnCpsServiceFlowTable
wmanIfCmnBsSsConfigurationTable

Figure 21. wmanIfCmnCps Structure

wmanIfCmnCps

This subtree contains objects germane to the CPS sublayer required by both the SS and the BS.

wmanIfCmnCpsServiceFlowTable

This table indexes a number of variables that pertain to the CPS layer of various Service Flows. Information such as MAC addresses, priority, tolerated jitter and latency, and Service Flow ID (SFID), are stored here, one row per Service Flow.

wmanIfCmnBsSsConfigurationTable

This table contains objects that define the timeouts and acceptable retry iterations for ranging requests as well as Automatic Repeat Requests (ARQ), as well as Dynamic Service Additions, Service Changes, and Service Deletions (DSA/DSC/DSD). This object resident on the SS must match the variable values on the BS with which it communicates.

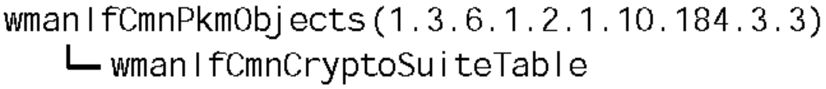
wmanIfCmnPkmObjects(1.3.6.1.2.1.10.184.3.3)
wmanIfCmnCryptoSuiteTable

Figure 22. wmanIfCmnPkmObjects Structure

wmanIfCmnPkmObjects

This subtree contains Security Sublayer objects shared between BS and SS.

wmanIfCmnCryptoSuiteTable

This lone PKM table object contains rows for the pairing of cryptographic methods used by the BS and SS. Each row contains values for the authentication method, the TEK algorithm, and the data encryption method.

B. WMANDEVMIB

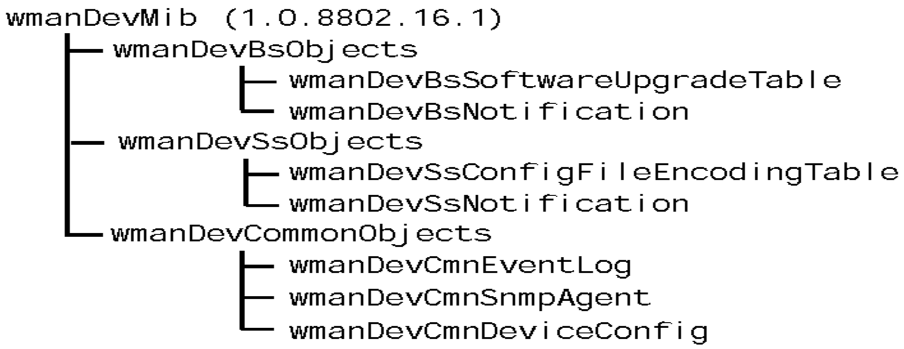

Figure 23. wmanDevMib Structure

This next set of MIB objects is a departure from the last group in that they are not associated with the interface between BS and SS. Every subtree and leaf within wmanIfMIB dealt with the storing and manipulation of variables that had a direct effect on the nature of the physical connection between BS and SS. This MIB, however, has no such objects, and concerns itself with system state variables for BS and SS alike. Another point of note is that the wmanIfMib and the wmanDevMib are under completely separate hierarchy chains under the SMI model. While wmanIfMib was under 1.3.6.1.2.1.10.184[1], wmanDevMib is under 1.0.8802.16.1.

1. wmanDevBsObjects

This subtree contains objects germane to the BS hardware and software.

wmanDevBsSoftwareUpgradeTable

This object table contains rows that identify information about the current running version of software on the BS. Information like Vendor ID, Software file name, and the timestamp of the last new software upgrade can all be found here.

[1] IANA has published an updated version of the SMI OID hierarchy in which the .184(propBWAp2Mp) subtree has been depreciated and replaced with the .237(ieee80216WMAN) subtree. However, the 802.16 Working Group has, as of this writing, not published an updated 802.16-2004 or 802.16f standard to reflect this change. As such, this thesis will refer to the depreciated .184(propBWAp2Mp) subtree for the sake of consistency given the use of the 802.16f standard.

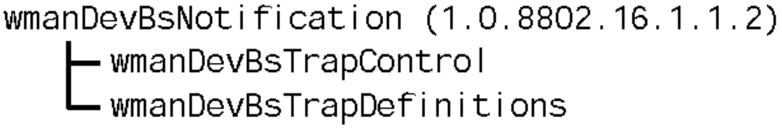

Figure 24. wmanDevBsNotification Structure

wmanDevBsNotification

This subtree contains objects germane to traps about the state of the BS sent to the NMS.

wmanDevBsTrapControl

This object defines what trap messages will be sent to the NMS. There are two options to set within the control, one allowing all of the notification types defined in the below object and another to inform the NMS that the number of logs currently exceeds the log buffer.

wmanDevBsTrapDefinitions

This object contains an index of all of the different types of notification that can be used by the BS.

2. wmanDevSsObjects

This subtree contains objects germane to the SS hardware and software.

wmanDevSsConfigFileEncodingTable

This table contains rows that store objects similar to that of the BS Software Update table. It contains hardware and software versions, as well as Vendor ID and configuration file parameters.[2]

[2] The lack of high level structure for wmanDevSsConfigFileEncodingTable was not an oversight. It is not included in the 802.16f standard.

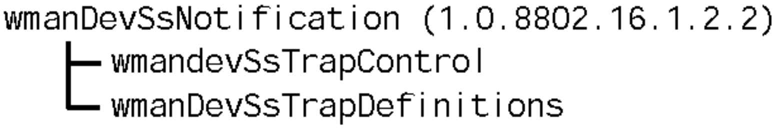

Figure 25. wmanDevSsNotification Structure

<u>wmanDevSsNotification</u>

This subtree contains objects germane to traps about the state of the SS sent to the NMS.

<u>wmanDevSsTrapControl</u>

This object defines what trap messages will be sent to the NMS. There are two options to set within the control, one allowing all of the notification types defined in the below object, and another to inform the NMS that the number of logs currently exceeds the log buffer.

<u>wmanDevBsTrapDefinitions</u>

This object contains an index of all of the different types of notification that can be used by the SS.

3. wmanDevCommonObjects

wmanDevCommonObjects

This subtree defines various event-based objects residing in both SS and BS.

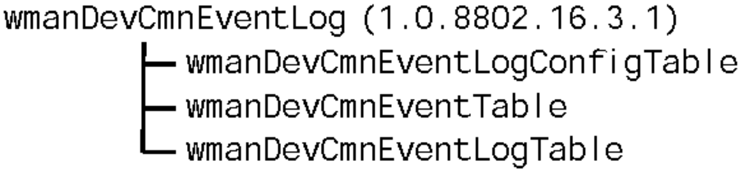
wmanDevCmnEventLog (1.0.8802.16.3.1)
wmanDevCmnEventLogConfigTable
wmanDevCmnEventTable
wmanDevCmnEventLogTable

Figure 26. wmanDevCmnEventLog Structure

wmanDevCmnEventLog

This subtree defines event log parameters for both BS and SS.

wmanDevCmnEventLogConfigTable

This object table contains rows for SS and BS. Each row contains values regarding the size of event logs, the severity of event to log, as well as log buffer sizes and the like.

wmanDevCmnEventTable

This table indexes the total number of events that can be used to create log messages in both BS and SS.

wmanDevCmnEventLogTable

This table is used to store local events. The standard states that "this table should reside in non-volatile memory that should presist(sic) after power cycle or reboot [16]."

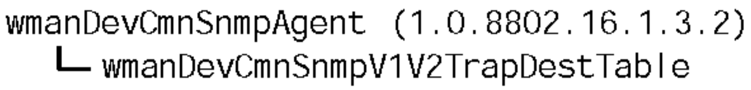
wmanDevCmnSnmpAgent (1.0.8802.16.1.3.2)
wmanDevCmnSnmpV1V2TrapDestTable

Figure 27. wmanDevCmnSnmpAgent Structure

wmanDevCmnSnmpAgent

This subtree contains objects related to the implementation of an SNMP Agent on either BS or SS.

wmanDevCmnSnmpV1V2TrapDestTable

This table contains rows that define Agent parameters such as trap destinations and trap port number are stored here.

wmanDevCmnDeviceConfig

This object is defined by an enumerated INTEGER SYNTAX which, if set to 0, requires no action, but, if set to 1, will reset the device.

As is illustrated by the rather exhaustive list of objects in this chapter the MIB defined in the 802.16f standard can store any of hundreds of object variables relating to both the interface between the BS and SS, and the hardware and software used for the BS and SS, respectively. The listing of the MIB variables is important, as it provides the basis for the ultimate purpose of this thesis. If there is a way to use SNMP to create a BS to SS session that wouldn’t require the SS to constantly respond, it will be found in these MIB objects.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESEARCH

Now that the foundations of IEEE Std. 802.16-2004, SNMP, and the IEEE Std. 802.16f have been established, the nature of the thesis can be fully grasped. The question again is whether or not SNMP can be used to create an IEEE Std. 802.16 session between BS and SS that does not require the SS to actually be involved in the conversation. The ultimate answer to the question rests on whether or not SNMP can be used to inject, using a series of SET commands, information into the BS that would allow it to believe that it had an established session with an SS of our choosing. This faux session could then be used to fool the BS into sending the SS information without requiring the SS to respond at all.

A. METHODS

The first step to establishing this possibility is to look back to the MIB and see what is and what is not of use. Of the two MIB modules, wmanIfMib and wmanDevMib, only the Interface MIB contains values linked to the establishment and use of a communications session. Thusly, the wmanDevMib can be ignored going forward, as it adds no values that can be exploited toward the end goal.

The second step is to focus on those connection state variables present in a BS implementation, and that are not reflected in the wmanIfMib. All told, there are about seven hundred and fifty objects in the 802.16f standard, most of which are found in the wmanIfMib module. To draw each one out, one-by-one for comparison, would be exhaustive and ultimately futile. Due to the robust nature of the 802.16f standard, one will find that the MIB has numerous instances where the connection state variables and the MIB objects are aligned. As such, the remainder of this chapter will focus on which state variables are absent from the MIB. To give structure to this conversation it may be helpful to return to the previous discussion detailing the setup process the SS goes through when initially logging on to a network.

As before, this discussion will start with power up of the SS, assuming that a BS is already setup to serve the geographic area for our hypothetical SS. Herein lies the first of the reasons why the aim of this thesis cannot be met by SNMP alone. The default behavior of an SS initially is to listen for a BS transmitting signals it can parse. If it picks up a signal from a BS containing a UL/DL-Map the first thing it will do is synchronize to the UL-Map and attempt to make contact with the BS. Worse yet, it will try to contact the BS starting out with the lowest power setting. This potentially means that the SS will broadcast multiple times until it reaches a power level that the BS can receive clearly. In fact, the whole of the start up process causes problems as the default behavior of the SS is to establish a connection first and take instruction later.

The IEEE Std. 802.16 BS and SS standards were designed such that, out of the box, an SS would attempt to create a connection with a local BS. This process requires a number of transmissions back and forth setting up everything from security keys to channel description information. This is of great benefit in the commercial sector as it significantly reduces the need for on-site support to any user standing up a SS. However, for the purpose of this thesis, it is a non-trivial hurdle in that this behavior antedates SNMP’s ability to manage the asset. As mentioned previously, SNMP is a layer 7 networking protocol. The initial creation of the BS to SS connection starts at layer 1 with the negotiation of PHY characteristics, then moves on to layer 2 properties such as frames, and then layer 3 variables such as IP addresses and such. In fact, it is only after every dimension of the BS to SS dynamic has been negotiated and set, that SNMP can even enter the connection to manage the assets. This fact alone prevents a truly silent SS. However, even if this were not the case, there are several other aspects of the IEEE Std. 802.16 to 802.16f comparison where a silent SS could not work.

Suspending the above complication, another significant area of difficulty is the authentication of the SS. Remember that, for an SS to gain access to the network, it needs to authenticate to the BS, which is done through the transmission of X.509 certificates. Obviously, if an SS is not transmitting, it cannot transmit the appropriate certificates. This in and of itself is expected. The desire is to use SNMP to inject information into the BS, which creates the illusion of an active SS; however, compounding the problem is the fact that the BS MIB does not even allow setting of an X.509 variable.

In the half of wmanIfMib dedicated to the SS there is an object, wmanIfSsDeviceCertTable. This object is a sequence of object wmanIfSsDeviceCertEntry which, for each SS, holds objects wmanIfSsDeviceCert and wmanIfSsDeviceManufCert. These objects contain the encoded X.509 certificate for both the SS and the CA root signed manufacturers certificate.

```
--
-- Table wmanIfSsDeviceCertTable
--
wmanIfSsDeviceCertTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF       WmanIfSsDeviceCertEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
            "This table describes the PKM device certificates for each
            SS wireless interface."
      ::= { wmanIfSsPkmObjects 3 }

wmanIfSsDeviceCertEntry OBJECT-TYPE
      SYNTAX      WmanIfSsDeviceCertEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
            "Each entry contains the device certificate of one SS."
INDEX { ifIndex }
::= { wmanIfSsDeviceCertTable 1 }

WmanIfSsDeviceCertEntry ::= SEQUENCE {
      wmanIfSsDeviceCert                                OCTET STRING,
      wmanIfSsDeviceManufCert                           OCTET STRING}


wmanIfSsDeviceCert OBJECT-TYPE
      SYNTAX      OCTET STRING (SIZE(0..65535))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
            "The X509 DER-encoded subscriber station certificate."
::= { wmanIfSsDeviceCertEntry 1 }

wmanIfSsDeviceManufCert OBJECT-TYPE
      SYNTAX      OCTET STRING (SIZE(0..65535))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
            "The X509 DER-encoded manufacturer certificate which is
      signed by the CA root authority certificate."
::= { wmanIfSsDeviceCertEntry 2 }
```

However, while there is an SS MIB object to contain the local X.509 certificates, there is no corresponding BS object which contains the certificates of all connected SS. The next step was to look for an object that would list SS PKM status as either accepted or rejected in the hope of being able to set the value to accepted, therefore bypassing the actual certificate exchange. Fortunately there is such an object type; unfortunately, its access type is “read-only,” so no SET messages could be sent to it.

```
wmanIfBsSsPkmAuthValidStatus OBJECT-TYPE
      SYNTAX      INTEGER {unknown (0),
                        validSsChained (1),
                        validSsTrusted (2),
                        invalidSsUntrusted (3),
                        invalidCAUntrusted (4),
                        invalidSsOther (5),
                        invalidCAOther (6)}
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "Contains the reason why a SS's certificate is deemed valid
            or invalid. Return unknown if the SS is running PKM mode.
            ValidSsChained means the certificate is valid because it
            chains to a valid certificate. ValidSsTrusted means the
            certificate is valid because it has been provisioned to be
            trusted. InvalidSsUntrusted means the certificate is
            invalid because it has been provisioned to be untrusted.
            InvalidCAUntrusted means the certificate is invalid
            because it chains to an untrusted certificate.
            InvalidSsOther and InvalidCAOther refer to errors in
            parsing, validity periods, etc, which are attributable to
            the SS certificate or its chain respectively."
      ::= { wmanIfBsSsPkmAuthEntry 17 }
```

As it turns out, SNMP doesn’t buy much, as the SS by definition transmits all of the connection negotiation information before SNMP can even be used. It also turns out that the BS isn’t all that much help either, as even if information were injected into the BS via SNMP to create a phantom SS, the PKM variables needed are read-only which prevents an NMS from being able to configure those values.

Given that SNMP is only an option after the fact of connection establishment; a truly silent SS doesn’t seem to be a viable option using SNMP as a mitigating method. However, another possibility would be for an SS to establish a session and then go silent.

In this amended situation, an SS would power up and proceed to initiate communication with a local BS; however, instead of continuing to send and receive normally, the SS would then cease transmission and the connections already provisioned would be held open in the absence of continued SS communication. A caveat that immediately needs to be addressed is that there would need to be some way of making the SS cease communications while keeping provisioned sessions open as well. While the possibility of keeping provisioned connections open via SNMP remains, forcing an

SS to cease communications with a BS does not. As applications on the SS request network services, there is currently no way in the IEEE Std. 802.16 standard to prevent those requests from being broadcast short of disabling the radio.

Given the above caveat, the key to enabling a silent SS after initial negotiation would be to artificially manipulate the timers associated with connections such that they wouldn't time out. For example, when an SS establishes a connection with a BS, the MIB variable wmanIfBsSsPkmAuthLifetime is set.

```
wmanIfBsSsPkmAuthLifetime OBJECT-TYPE
      SYNTAX      Integer32 (86400..6048000)
      UNITS       "seconds"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
            "The value of this object is the lifetime, in seconds, the
            BS assigns to an authorization key for this SS."
      REFERENCE
            "Table 341 in IEEE Std 802.16-2004"
      DEFVAL      { 604800 }
      ::= { wmanIfBsSsPkmAuthEntry 5 }
```

Notice that the value of the above object is in seconds and that it can be between 86,400 and 6,048,000 with a default value of 604,800. This means that, at its maximum, this object would create a 70-day timer for an SS authentication key. Since this upper limit is well beyond the battery life of any handheld device capable of hosting an SS, this would create the necessary space for the amended silent SS.

This process, however, has no need of SNMP to configure. The BS configuration file allows an administrator to set the initial key lifetimes (for both AK and TEK).
"After an SS completes basic capabilities negotiation, it shall initiate an authorization exchange with its BS. The BS's first receipt of an Auth Request message from the unauthorized SS shall initiate the activation of a new AK, which the BS sends back to the requesting SS in an Auth Reply message. This AK shall remain active unit lit expires according to its predefined *AK Lifetime*, a BS system configuration parameter." Page 298 802.16-2004 Std

Thus far it has been shown that a silent SS is a practical impossibility under the current architecture of IEEE Std. 802.16. On initial start up, the IEEE Std. 802.16 SS, upon receiving a DCD message and DL-Map, will broadcast back to the BS over and over again until a connection is established. Once the connection is established, all seven layers of the OSI stack must be built by multiple negotiation transmissions from SS and BS both. Only after all of this default behavior could SNMP even be considered, as it is an application layer protocol. Even if the transmitter on the SS could be disabled such that it could still receive without transmitting, there is currently no combination of SNMP SET variable that can be sent to a BS such that it would transmit to an SS with which it had not performed initial negotiations. Finally, were an SS to negotiate an initial connection and then stop transmitting, there is a possibility of setting overlong timeouts to ensure a continual connection; however, this capability doesn't require SNMP to be set, as it is a standard configuration option of an IEEE Std. 802.16 BS. While the SNMP approach fails to meet the goals of this thesis, in researching IEEE Std. 802.16 capabilities and functionality, another possibility for creating a silent SS implementation has become apparent. To describe this alternative method, a look back to the use case example of soldiers surrounding a FOB is warranted.

B. ALTERNATIVE SILENT SS

Earlier in this thesis, a FOB use case was crafted to give a framework in which to discuss issues relevant to the goal of creating a silent SS. The need for a stable, robust wireless network was identified as a core component, and the distance, throughput, reliability, and minimum setup overhead of IEEE Std. 802.16 were shown to make it an ideal technology for supporting such use cases.

It was also identified that concealment was a key component of the mission of these forward forces, and, while IEEE Std. 802.16 is a great boon for NCW, its default transmission behavior could be used against the soldier using IEEE Std. 802.16 equipment. The availability of tools which can detect and locate the source of radio transmissions creates an unacceptable risk to forward forces who rely on concealment, yet these very forces are the ones that stand to gain the most from NCW, as changing COP information is likely to affect them first.

So, the question becomes one of how to provide these forces with all the gains of NCW enabled equipment without incurring the risk of exposure due to transmission detection. This thesis endeavored to determine whether SNMP could provide this capability. The default behavior of IEEE Std. 802.16 SS and the values of the IEEE Std. 802.16f MIB have shown SNMP cannot currently meet this goal today. Given that SNMP cannot support the goal, are there other options?

Here it might be useful to review the requirements of such a network connection.

- The connection should support network connectivity from the FOB to units near and far.
- The connection should be able to be configured such that it supports the concealment of those units who rely on concealment (i.e., no transmissions).
- While not transmitting, the connection should still be able to receive information broadcast to it allowing hidden units to benefit from NCW enabled equipment.
- The information going out need not relate specifically to the end user. A generic subset of COP information should be dispersed as widely as possible such that all users are reachable.

While the SNMP route failed to provide the functionality desired, could there be other options for meeting the above requirements? Another possibility lies in using a proxy SS to communicate on behalf of all the silent SS and transmit to them in a broadcast fashion.

In this new situation, the BS functions exactly as before. However, there is one fully functional SS communicating with the BS, which is acting as a proxy for all other BS. In this situation the proxy SS would request to send the above mentioned generic COP information to a broadcast address at layer two of the OSI model. At layer two, each device on the network has a unique address (MAC Address) to identify it from any other host on the network. However, if information is sent to a special broadcast address that data is sent to all hosts on the local network. Additionally, whereas individual network hosts listen for information and ignore all but what is specifically addressed to them, broadcast traffic is accepted and processed by every end node in the network. The proxy SS would have to insert itself as a session peer for the network traffic sent from it to be broadcast to every other silent SS. In this way, the proxy SS could constantly request that COP information be sent to a broadcast address at which point the BS would then broadcast out the message to all SS within range. The SS in this situation would correctly interpret said traffic as something it should try to process and pass the broadcast frame up the OSI stack to the application layer.

This approach has the advantage of needing no additional configurations/modifications on the part of the BS. Furthermore, since the broadcast frames will be received just like any other traffic, there is no need to manipulate any layer of the OSI stack to get the SS to accept and process the traffic as it is received. It should be mentioned that this theory has not been borne out by research like the question of SNMP, however, it is a plausible next step in creating space for a silent SS to function in an IEEE Std. 802.16 implementation.

VI. CONCLUSION

This thesis has analyzed IEEE Std. 802.16, SNMP, and IEEE Std. 802.16f in an attempt to discern whether or not SNMP could be used to remotely configure an IEEE Std. 802.16 Subscriber Station (SS) to receive data while not transmitting in return. As modern military forces seek to exploit the gains of Network Centric Warfare (NCW), the question of the best network technology to use becomes an important one. IEEE Std. 802.16 (WiMAX) has shown itself to be an optimal choice for extending the reach of the internet down to the level of the individual soldier. The range, throughput, flexibility, and ability to function both at the edge and at the core of a network, combined with its robust scheduling MAC which remains stable under overload conditions, make IEEE Std. 802.16 an ideal choice for pushing networking power to the edges of modern battlespaces. However, for all its advantages, the default transmit/receive nature of IEEE Std. 802.16 equipment could be exploited by enemy forces and used to pinpoint the location of troops equipped with IEEE Std. 802.16 gear. This thesis focused on the use of SNMP in combination with IEEE Std. 802.16 to mitigate said risk while still allowing IEEE Std. 802.16 equipped soldiers to download important data. The research into the question of creating a silent SS began in chapter II with the IEEE 802.16 standard.

The IEEE 802.16 standard functions at the two lowest layers of the 7 Layer OSI reference model. The 802.16 standard itself is further subdivided in to the additional layers of:

- The Service Specific Convergence Sublayer (CS) which receives traffic from various Layer 3 protocols and formats it such that the next lower layer can become “protocol agnostic.”
- The Common Part Sublayer (CPS) which is responsible for the core MAC functions of IEEE Std. 802.16 such as medium access, connection management, and QoS functions.
- The Privacy Sublayer which authenticates SS to BS to prevent TOS/DOS and encrypts payload data between BS and SS.

- The PHY layer which corresponds to layer 1 of the OSI model and deals with the means and method of getting signals out into the air.

These layers are implemented by the two IEEE Std. 802.16 entities, the BS and the SS. The BS is the central aggregation point for all traffic on the network. All SS must go through the BS, and obey its rules for connection, to transmit data anywhere. The BS schedules all of the traffic coming from and going to all SS. Other than a brief window of contention during SS initial sign on, the connections between BS and SS are contention free, which allows IEEE Std. 802.16 links to enjoy a very high level of communications efficiency as compared to other wireless technologies. In chapter III, SNMP was analyzed as to how it could be used to create a silent SS.

SNMP has been shown through the use of the FCAPS model to be an extremely useful tool in managing today's modern network. Through the use of its lightweight proven communication scheme, SNMP lends itself well to supporting high availability through fault detection and performance management. By way of the SNMP Management Information Base (MIB), it has been shown that SNMP can be used to remotely configure networking assets as well as monitor those assets for usage, faults, and anomalies. As the IEEE 802.16 Working Group has released the default MIB for the 802.16 standard (known as IEEE Std. 802.16f), chapter IV centered on what MIB objects could be modified or otherwise manipulated to aid in creating a silent SS.

The IEEE Std. 802.16 MIB contains over a thousand objects related to the implementation of IEEE Std. 802.16. These objects are broadly broken up into two MIB modules; wmanIfMib describes the nature of the interface between BS and SS, and wmanDevMib describes objects related to the specific instances of a BS or SS. However, while the nature of the IEEE Std. 802.16f MIB is robust in chapter V, it was shown to be inadequate in configuring an SS to be silent.

The use of SNMP and the intersection of IEEE Std. 802.16 and SNMP, embodied in the IEEE 802.16f standard, came up short in creating a silent SS in two primary ways.

- The default behavior of an SS on power up is to transmit in order to locate and connect with a local BS. SNMP is an application and as such resides at layer 7 in the OSI model. When an SS does establish a connection with a BS, it builds that connection layer by layer, so, by the time SNMP configuration is even an option, the SS has transmitted back and forth to the BS numerous times, establishing layers 1 thru 6.

- Even if it were possible to mute the default transmissions of a SS, there is no combination of SET commands one could issue using SNMP to trick a BS into thinking it had an authenticated connection with an SS it had never actually heard from. This is most clearly illustrated by the fact that IEEE Std. 802.16 uses X.509 certificates to authenticate SS to BS, and in the IEEE Std. 802.16f MIB there are no objects to store a phantom certificate.

A third approach to the problem would be to use SNMP to create overlong timeouts such that an SS could establish a connection with a local BS and then go silent. The overlong timeouts would then allow the BS to continue transmission even though the SS has stopped transmitting. However, this approach does not require SNMP at all as the timeout setting is a regular configuration setting an administrator would set upon standing up a BS anyway. Unfortunately, the research bore out that there was no way to use the current combination of IEEE Std. 802.16 and SNMP to create a silent SS, however in the research another option came to light which may hold potential for meeting the ends of this thesis.

It was discovered that when a BS sends out initial connection information, it does so in a way such that SS who are not yet connected can receive it. This broadcast traffic can be seen and interpreted by all SS within range of the BS regardless of whether they have attempted to connect to the BS or not. Therefore, it is possible to send traffic to an SS without that SS having established a prior connection with the BS. The possible implementation of this theory seeks to use the nature of broadcast addressing to send traffic from a BS to a broadcast address that every SS would be capable of receiving and processing regardless of their connection state with the BS. In this implementation, one SS would establish a connection with the BS and act as a proxy on behalf of all local silent SS, requesting pertinent data on their behalf. This theory however, has not been tested like the SNMP approach and remains only a theory.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. FURTHER RESEARCH

This thesis focused on using SNMP in combination with a default IEEE Std. 802.16 implementation to create a connection between an SS and BS where the SS could remain silent while still benefitting from the transmissions of the BS. While this proved impossible under current conditions, several other possibilities became apparent during the research. The following section briefly discusses several alternative options worth further research.

A. PROXY SS

A possibility exists for creating a silent SS situation where a single proxy SS, fully connected to the local BS, transmits and receives on behalf of the number of silent SS in the area. This approach would have to take advantage of the multicast and broadcast capabilities of IEEE Std. 802.16. While unicast messages are sent from one sender to one receiver, multicast (one to many) and broadcast (one to all) are special addressing methods which allow a sender to send out only one copy of data to be received by either a select number or all other hosts on the network.

In a unicast transmission, data is sent from the address of the sender to the address of the receiver. If the sender needs to send data to a number of hosts, the overhead incurred by sending out copy after copy to each individual receiver can be a significant cause of network saturation. However, by using broadcast/multicast addresses, a sender can transmit only one copy of the data, and the network infrastructure will take care of the rest. In the case of multicast, only one copy per transmission is necessary to ensure that all interested receivers get the data, and likewise in the case of broadcast, to ensure that all hosts on the network get a copy of the data.

Drawing upon the FOB scenario, a proxy SS could be set up within range of a BS, and would constantly request COP data from the Wide Area Network (WAN). Upon receipt, this proxy SS would then send the data to a multicast/broadcast address. The forwarded COP traffic would then go back to the BS which would then be multicast/broadcast out to all silent SS. The proxy SS in this scenario is necessary as there needs to be at least one node requesting traffic from the BS in order for traffic to be forwarded to the BS.

This situation does, however, raise a number of questions about the process of the proxy SS concept.

- Which method, broadcast or multicast, would provide the broadest dissemination of data to N number of potential silent SS while allowing for the least network saturation?
- As there are broadcast Layer 3 addresses (IP addresses) and broadcast Layer 2 addresses (MAC addresses), what gains are associated with using the broadcast address of one Layer over the other?
- Since it has been established that the default behavior of the SS is to transmit, how could this behavior be changed such that an SS would be a passive receiver?
- In the proxy SS example, the proxy SS is close to the BS, meaning they would negotiate a modulation setting appropriate for nearby BS and SS. However, the goal would be to service SS that are at the far reaches of the BS range. How could the BS be configured to broadcast data using a modulation which favors range over throughput, when the proxy SS is using a modulation which favors throughput over range?

B. LPI PHY

As has been shown, the IEEE Std. 802.16 PHY layer can be modified again and again to meet the ever-changing needs of the consumer. So long as the new PHY conforms to the Layer1/Layer2 SAP, new PHY layers can be created arbitrarily. One possibility, which would obviate the need for a silent SS, but would still mitigate the risk of exposure due to transmission tracking, is that of a LPI PHY. With a LPI PHY the BS and SS could still communicate with each other in full duplex fashion, however, since the PHY method is LPI there is no additional risk associated in transmitting back and forth.

An example scenario would be if the LPI PHY developed for IEEE Std. 802.16 worked like Code Division Multiple Access (CDMA). In CDMA the transmitter multiplies the intended transmission signal by a pseudo-noise code which spreads the energy of the signal over a wider band. This spread out signal, when transmitted, looks to the outside observer (in fact to anyone without the pseudo-noise code) as mere background noise. Upon receipt, the known pseudo-noise code can be used to recover the original signal.

This LPI method would be of significant benefit as it provides gains not only in the protection of soldiers, but in the protection of data. It should be noted that the SS usage of the lowest power settings for transmission to the BS is already a good LPI measure, however it still lends itself to detection if a tracking device is close enough. If a true LPI measure is employed, then the questions associated with Layer 2 security become moot. If the signal can only be received by the indented recipient then the LPI approach adds data confidentiality as well as physical exposure protection all in one modification.

However, while the LPI method does hold promise, a number of questions arise out of its proposed use.

- An LPI approach like CDMA would be contention based. As one of the gains associated with IEEE Std. 802.16, and perhaps the most significant, is its scheduling MAC layer, which is stable under overload conditions, the question then becomes, “Is there a way to implement LPI technology while keeping the scheduling MAC and contention free connections?”
- As the pseudo-noise code used in CDMA spreads the signal over a much wider band, would the modulation techniques used in IEEE Std. 802.16 be affected by changes in signal energy?
- If a LPI PHY were implemented in IEEE Std. 802.16, what would the effect be on the ranges and throughput of the NLOS connections? The LOS connections?

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Wikipedia, “Metcalfe’s law,” http://en.wikipedia.org/wiki/Metcalfe's_law, March 2008.

[2] Joint Pub 3-13, www.dtic.mil/doctrine/jel/new_**pub**s/jp3_13.pdf, March 2008.

[3] Sprint, Inc., “Sprint Nextel Cites WiMAX Network Progress For 2007,” http://www.networkworld.com/news/2006/080806-sprint-nextel-IEEE Std. 802.16 .*html*?prl?ap1=rcb, March 2008.

[4] W. M. Hall, Vantage Point, “Military Intelligence Professional Bulletin,” http://www.fas.org/irp/agency/army/tradoc/usaic/mipb/1998-4/vantage.htm, June 2008.

[5] D. S. Alberts, J. Garstka, F. P. Stein, *Network Centric Warfare: Developing and Leveraging Information Superiority*, National Defense University Press, 1999

[6] United States, “Defense Acquisitions Restructured JTRS Program Reduces Risk, but Significant Challenges Remain: Report to Congressional Committees,” 2006, [Washington, D.C.]: U.S. Government Accountability Office. http://www.gao.gov/new.items/d06955.pdf, April 2008.

[7] C. Eklund, *WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Area Networks,* IEEE Press, 2006

[8] D. Comer, *Computer Networks and Internets*. Saddle River, N.J.: Prentice Hall, 1999.

[9] M. Norton, “The Scholar’s Approach to the Internet,” http://www.oreillynet.com/pub/a/network/2001/02/09/net_2nd_lang.html, 2001, August 2008.

[10] H. Yin, S. Alamouti, "OFDMA: A Broadband Wireless Access Technology," in *Sarnoff Symposium, 2006 IEEE* , vol., no., pp.1-4, 27-28, March 2006.

[11] F. Ohrtman, *IEEE Std. 802.16 Handbook: Building 802.16 Wireless Networks*, McGraw-Hill Communications. New York: McGraw-Hill, 2005.

[12] S. J. Harnedy, *Total SNMP: Exploring the Simple Network Management Protocol*. Upper Saddle River, NJ: Prentice Hall PTR. 1998.

[13] D. R. Mauro, K. J. Schmidt, *Essential SNMP*, Beijing: O'Reilly, 2005.

[14] D. Perkins, E. McGinnis, *Understanding SNMP MIBs*, Upper Saddle River, N.J.: Prentice Hall PTR, 1997.

[15] K. McCloghrie, M. Rose, IETF, “Management Information Base for Network Management of TCP/IP-based internets: MIB-II,” 1991, http://tools.ietf.org/html/rfc1213, August 2008.

[16] IEEE Computer Society, IEEE Microwave Theory and Techniques Society, IEEE Standards Board, Institute of Electrical and Electronics Engineers, & IEEE Xplore (Online service). “IEEE standard for local and metropolitan area networks. amendment 1 : management information base Part 16, Air interface for fixed broadband wireless access systems.” 2005, http://ieeexplore.ieee.org/servlet/opac?punumber=10438, March 2008.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Know Library
   Naval Postgraduate School
   Monterey, California

3. Dan Boger
   Naval Postgraduate School
   Monterey, California

4. Rex Buddenberg
   Naval Postgraduate School
   Monterey, California

5. Albert Barreto
   Naval Postgraduate School
   Monterey, California